



工业和信息产业科技与教育专著出版资金资助出版  
基于岗位职业能力培养的高职网络技术专业系列教材建设

# Linux系统管理教程

许兴鹏 黄道金 主 编

简庆龙 黄君羨 副主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING



## 内 容 简 介

本书围绕 Linux 系统管理员、系统工程师等岗位对 Linux 基础核心技能的要求，以主流的 Red Hat Enterprise Linux 6.x 系统平台为载体，采用任务驱动模式编写。

本书主要内容包括：认识 Linux、安装 Linux、GNOME 图形桌面使用入门、Bash 入门、vim 编辑器、管理用户和组、Linux 文件权限管理、建立网络连接、管理物理存储和建立文件系统、逻辑卷的使用、安装和管理软件包、创建计划任务、监控系统资源和运行状态及系统日志分析。最后，本书还包含两个附录以便读者快速参考，分别是 VMware Workstation 使用入门和 Bash 常用基础命令用法示例。

本书可作为高职院校网络专业及相关专业教材，也适合作为社会培训和系统管理人员的参考用书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

## 图书在版编目（CIP）数据

Linux 系统管理教程 / 许兴鹏，黄道金主编. —北京：电子工业出版社，2015.8

基于岗位职业能力培养的高职网络技术专业系列教材建设

ISBN 978-7-121-26781-9

I. ①L… II. ①许… ②黄… III. ①Linux 操作系统—高等职业教育—教材 IV. ①TP316.89

中国版本图书馆CIP数据核字（2015）第171272号

策划编辑：束传政

责任编辑：束传政

特约编辑：赵海红 罗树利

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编：100036

开 本：787×1092 1/16 印张：13.75 字数：347千字

版 次：2015年8月第1版

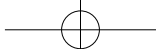
印 次：2015年8月第1次印刷

印 数：3000册 定价：35.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。



## 编委会名单

### 编委会主任

吴教育 教授 阳江职业技术学院院长

### 编委会副主任

谢赞福 教授 广东技术师范学院计算机科学学院副院长

王世杰 教授 广州现代信息工程职业技术学院信息工程系主任

### 编委会执行主编

石 硕 教授 广东轻工职业技术学院计算机工程系

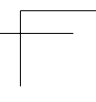
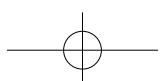
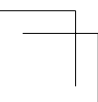
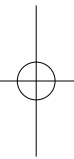
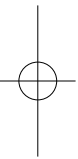
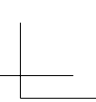
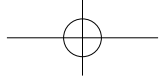
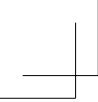
郭庚麒 教授 广东交通职业技术学院人事处处长

### 委员（排名不分先后）

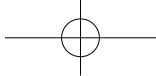
王树勇	教授	广东水利电力职业技术学院教务处处长
张蒲生	教授	广东轻工职业技术学院计算机工程系
杨志伟	副教授	广东交通职业技术学院计算机工程学院院长
黄君羨	微软认证专家	广东交通职业技术学院计算机工程学院网络工程系主任
邹 月	副教授	广东科贸职业学院信息工程系主任
卢智勇	副教授	广东机电职业技术学院信息工程学院院长
卓志宏	副教授	阳江职业技术学院计算机工程系主任
龙 翔	副教授	湖北生物科技职业学院信息传媒学院院长
邹利华	副教授	东莞职业技术学院计算机工程系副主任
赵艳玲	副教授	珠海城市职业技术学院电子信息工程学院副院长
周 程	高级工程师	增城康大职业技术学院计算机系副主任
刘力铭	项目管理师	广州城市职业学院信息技术系副主任
田 钧	副教授	佛山职业技术学院电子信息系副主任
王跃胜	副教授	广东轻工职业技术学院计算机工程系
黄世旭	高级工程师	广州国为信息科技有限公司副总经理

### 秘书

束传政 电子工业出版社 rawstone@126.com







## Preface

# 前言

作为最著名的自由和开源软件项目，Linux 是一种先进的操作系统，它支持更多的硬件平台，既可以运行在个人 PC 上，也可以运行在服务器或其他大型主机之上，如流行的网站后台；Linux 也广泛应用于嵌入式系统上，如手机、平板电脑、路由器、电视和电子游戏机等。在移动设备上广泛使用的 Android 操作系统就是创建在 Linux 内核之上的。

Linux 系统越来越流行，在企业服务器应用中更是如此。作为计算机专业的学生或 IT 从业人员，非常有必要了解和学习一些 Linux 基础知识。

本教程的编写旨在帮助 Linux 系统管理的初学者快速入门和提高。教程的设计紧跟 Linux 技术发展趋势，采用主流的 Red Hat Enterprise Linux 6.x 为演示系统，结合最新的虚拟化、云计算技术的应用，让学员全面了解 Linux 系统的知识；内容从熟悉 Linux 的图形界面开始到复杂的命令行，再到系统管理的综合应用，由浅入深，循序渐进，轻松入门。本教程在编写的过程中，特别注意对于学习者学习能力的提高，注重过程方法和解决问题，而不是死记硬背步骤；强调原理分析，但更加注重实践，教程中大量穿插其中的 REAL-WORK TIP 是作者多年 Linux 企业应用工作经验的总结。

为检验学习成果，教程每章节后面设置了一定的实践任务，分为两部分。

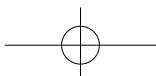
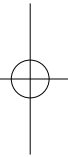
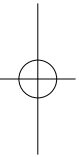
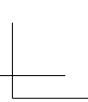
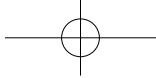
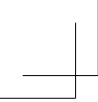
- 基础技能演练。对本章节的内容进行回顾练习，设置的任务对应章节中的知识点。所有问题都能从对应章节中找到答案或方法。主要是回顾和复习每节的重点内容。
- 提高技能实践。以章节讲解的知识点为基础，结合生产环境中的真实应用案例，形成综合性的实践任务。限于篇幅或难度一致性，题目中涉及的知识点可能在教材中并未提及，但通常教师会在教授的过程中进行扩展，这时请务必做好笔记。企业应用案例往往是综合性强、难度较高的，但也是最具实用价值的，并能很好地锻炼和提高学生的学习能力和分析、解决问题的能力。

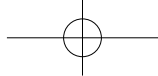
本教程由许兴鹄、黄道金担任主编，简庆龙、黄君羨担任副主编，徐务棠、李连天等同行参与部分章节的编写；最后由许兴鹄统稿。

由于时间仓促，加之作者水平有限，本教程难免存在疏漏和不足之处，欢迎指正。同时作者也会在技术应用和教学实践中反复提炼与修改，做到内容更加合理、实用。

作 者

2015 年 5 月





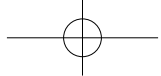
Contents

# 目录

<b>第1章 认识Linux.....</b>	<b>1</b>
1.1 历史 .....	2
1.2 Linux操作系统的基本结构 .....	5
1.3 Linux应用方向 .....	6
1.4 Linux的特性 .....	7
1.5 Linux的发行版 .....	7
1.6 Linux的版本 .....	9
1.7 其他 .....	10
本章总结.....	11
基础技能演练 .....	12
提高技能实践 .....	12
 <b>第2章 安装Linux.....</b>	 <b>13</b>
2.1 获取Red Hat企业版Linux.....	13
2.2 安装前的准备 .....	15
2.3 开始安装 .....	17
2.4 使用anaconda安装 .....	18
2.5 安装后的配置.....	32
2.6 高级安装选项 .....	36
2.7 删除Red Hat企业版Linux.....	38
本章总结.....	38
基础技能演练 .....	39
提高技能实践 .....	39



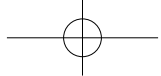
<b>第3章</b>	<b>GNOME图形桌面使用入门 .....</b>	<b>40</b>
3.1	登录到RHEL的GNOME桌面 .....	40
3.2	配置RHEL 6的显示分辨率和多显示器 .....	42
3.3	GNOME桌面概述 .....	43
3.4	结束当前会话 .....	46
3.5	文件管理操作 .....	47
3.6	通过gedit编辑文件 .....	51
3.7	使用配置工具查看和更改系统时间 .....	52
	本章总结 .....	54
	基础技能演练 .....	55
	提高技能实践 .....	55
<b>第4章</b>	<b>Bash入门 .....</b>	<b>56</b>
4.1	什么是Shell .....	56
4.2	使用Shell命令行 .....	56
4.3	在命令行中启动图形工具 .....	58
4.4	命令的格式 .....	59
4.5	Bash命令行的快捷键 .....	62
4.6	获得命令的帮助信息 .....	63
4.7	命令使用示例 .....	66
4.8	Bash命令行重要的高阶功能 .....	70
4.9	编写Bash脚本 .....	73
	本章总结 .....	73
	基础技能演练 .....	74
	提高技能实践 .....	75
<b>第5章</b>	<b>vim编辑器 .....</b>	<b>76</b>
5.1	在命令行中使用vim编辑器 .....	76
5.2	vim的3种模式 .....	77
5.3	退出vim编辑器 .....	78
5.4	vim命令模式下的常用操作 .....	78
5.5	vim末行模式下的常用操作 .....	80



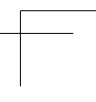
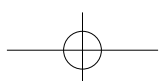
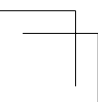
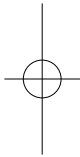
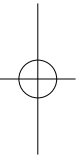
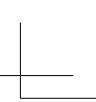
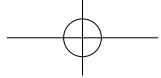
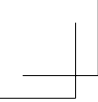
5.6 vim的其他操作 .....	81
本章总结 .....	82
基础技能演练 .....	82
提高技能实践 .....	83
<b>第6章 管理用户和组 .....</b>	<b>84</b>
6.1 使用图形环境用户管理工具 .....	85
6.2 使用命令行工具管理用户 .....	88
6.3 切换账号 .....	95
6.4 用户账号初始化 .....	96
本章总结 .....	98
基础技能演练 .....	99
提高技能实践 .....	99
<b>第7章 Linux文件权限管理 .....</b>	<b>101</b>
7.1 使用GUI工具管理权限 .....	102
7.2 在命令行中管理权限 .....	103
7.3 特殊权限 .....	105
7.4 隐藏的扩展属性（权限） .....	106
7.5 访问控制列表（ACL） .....	107
7.6 SELinux前瞻 .....	111
本章总结 .....	112
基础技能演练 .....	113
提高技能实践 .....	113
<b>第8章 建立网络连接 .....</b>	<b>114</b>
8.1 基本网络概念 .....	114
8.2 Linux的网络配置 .....	120
8.3 远程管理Linux系统初探 .....	125
本章总结 .....	128
基础技能演练 .....	129
提高技能实践 .....	129



<b>第9章 管理物理存储和建立文件系统.....</b>	<b>130</b>
9.1 硬盘的物理组成 .....	130
9.2 分区和文件系统的基本概念 .....	131
9.3 建立和使用磁盘分区 .....	132
本章总结 .....	136
基础技能演练 .....	136
提高技能实践 .....	136
 <b>第10章 逻辑卷的使用.....</b>	 <b>137</b>
10.1 逻辑卷的概念 .....	137
10.2 配置LVM的步骤 .....	138
10.3 扩大逻辑卷容量 .....	139
本章总结 .....	139
基础技能演练 .....	140
提高技能实践 .....	140
 <b>第11章 安装和管理软件包 .....</b>	 <b>141</b>
11.1 软件包管理介绍 .....	141
11.2 使用rpm命令安装和管理RPM包 .....	142
11.3 使用YUM工具安装和管理RPM包 .....	144
11.4 搭建自己的软件仓库 .....	147
本章总结 .....	147
基础技能演练 .....	148
提高技能实践 .....	148
 <b>第12章 创建计划任务.....</b>	 <b>150</b>
12.1 一次性计划任务 .....	150
12.2 周期性执行的计划任务 .....	151
本章总结 .....	153
基础技能演练 .....	153
提高技能实践 .....	154



<b>第13章 监控系统资源和运行状态</b>	<b>155</b>
13.1 查看和管理系统进程	155
13.2 收集系统运行状态信息	159
13.3 虚拟文件系统	160
本章总结	161
基础技能演练	161
提高技能实践	161
<b>第14章 系统日志分析</b>	<b>162</b>
14.1 查看系统日志	162
14.2 日志产生机制	163
14.3 日志切割与轮替	165
本章总结	166
基础技能演练	166
提高技能实践	167
<b>附录A VMware Workstation使用入门</b>	<b>168</b>
A.1 什么是 VMware Workstation	168
A.2 软硬件需求	168
A.3 安装和使用VMware Workstation	170
A.4 创建虚拟机	173
A.5 使用虚拟机	182
<b>附录B Bash常用基础命令用法示例</b>	<b>189</b>
B.1 文件和目录相关的命令	189
B.2 显示和查看相关的命令	194
B.3 查找相关的命令	196
B.4 查看系统信息相关的命令	199
B.5 系统管理相关命令	203
B.6 打包压缩相关的命令	205
<b>参考文献</b>	<b>208</b>







# 第1章

## 认识Linux

Linux (['linəks][□linəks])<sup>1</sup> 是一种自由和开放源代码的类 UNIX 操作系统。该操作系统的内核由林纳斯·托瓦兹 (Linus Torvalds) 在 1991 年 10 月 5 日首次发布, 再加上用户空间的应用程序之后, 成为 Linux 操作系统。Linux 也是自由软件和开放源代码软件发展中最著名的例子。只要遵循 GNU 通用公共许可证, 任何个人和机构都可以自由地使用 Linux 的所有底层源代码, 也可以自由地修改和再发布。大多数 Linux 系统还包括了像提供 GUI 界面的 X Window 之类的程序。除了一部分专家之外, 大多数人都是直接使用 Linux 发布版, 而不是自己选择每一样组件或自行设置。

严格来讲, 术语 Linux 只表示操作系统内核本身, 但通常采用 Linux 内核来表达该含义。Linux 则常用来指基于 Linux 内核的完整操作系统, 包括 GUI 组件和许多其他实用工具。由于这些支持用户空间的系统工具和库主要由理查德·斯托曼 (Richard M. Stallman) 于 1983 年发起的 GNU 计划提供, 自由软件基金会提议将该组合系统命名为 GNU/Linux, 但 Linux 本身不属于 GNU 计划。

Linux 最初是作为支持英特尔 x86 架构的个人计算机的一个自由操作系统。目前 Linux 已经被移植到更多的计算机硬件平台, 远远超出其他任何操作系统。Linux 是一个领先的操作系统, 可以运行在服务器和其他大型平台之上, 如大型主机和超级计算机。世界上 500 个最快的超级计算机 90% 以上运行 Linux 发行版或变种, 包括最快的前 10 名超级计算机运行的都是基于 Linux 内核的操作系统。Linux 也广泛应用在嵌入式系统上, 如手机、平板电脑、路由器、电视和电子游戏机等。在移动设备上广泛使用的 Android 操作系统就是创建在 Linux 内核之上的。

通常情况下, Linux 被打包成供个人计算机和服务器使用的 Linux 发行版, 一些流行的主流 Linux 发布版, 包括 Debian (及其派生版本 Ubuntu、Linux Mint)、Fedora (及其相关版本 Red Hat Enterprise Linux、CentOS) 和 openSUSE 等。Linux 发行版包含 Linux 内核和支撑内核的实用程序和库, 通常还带有大量可以满足各类需求的应用程序。个人计算机使用的

<sup>1</sup> 本章节部分内容来自维基百科和其他互联网内容 (如发行版官网介绍等)。



Linux 发行版通常包 X Window 和一个相应的桌面环境，如 GNOME 或 KDE。桌面 Linux 操作系统常用的应用程序，包括 Firefox 网页浏览器、LibreOffice 办公软件、GIMP 图像处理工具等。由于 Linux 是自由软件，任何人都可以创建一个符合自己需求的 Linux 发行版。

## 1.1 历史

### 1. 未完成的 Multics

早期的计算机并不像现在的微型 PC，随处可见，它们只出现在军事、科研和教育等领域，并且为数不多的计算机不仅慢还很难使用。20 世纪 60 年代初期，麻省理工大学（MIT）开发了“兼容分时系统（Compatible Time-Sharing System, CTSS）”，可以让大型机通过多个终端（Terminal）联机进入使用主机资源。1965 年前后，由美国电话及电报公司（AT&T）贝尔实验室、麻省理工大学（MIT）及通用电气公司（GE）计划开发一个多用途（General-Purpose）、分时（Time-Sharing）及多用户（Multi-User）的操作系统，也就是这个 Multics（MULTiplexed Information and Computing System），其被设计运行在 GE-645 大型主机上。不过，这个项目由于太过复杂，整个目标过于庞大，糅合了太多的特性，进展太慢，Multics 虽然发布了一些产品，但是性能都很低，于是到了 1969 年 2 月，AT&T 最终撤出了投入 Multics 项目的资源，终止了这项合作计划。不可否认，Multics 系统是一个优秀的设计，后面出现的 UNIX 系统一定程度上受到它的启发。

### 2. UNIX 和 BSD

我们通常说 Linux 是一个 UNIX- Like（类 UNIX）操作系统，继承了 UNIX 高效、稳定、安全的特性，并与 UNIX 保持着高度兼容性，我们常用的 Linux 系统整合着大量原本在 UNIX 下的工具与服务。

UNIX 操作系统是美国 AT&T 公司贝尔实验室于 1969 年实现的操作系统。最早由肯·汤普逊（Ken Thompson）、丹尼斯·里奇（Dennis Ritchie）、道格拉斯·麦克罗伊（Douglas McIlroy）和乔伊·欧桑纳（Joe Ossanna）于 1969 年在 AT&T 贝尔实验室开发，于 1971 年首次发布，最初完全用汇编语言编写，这是当时的一种普遍的做法。后来，在 1973 年用一个重要的开拓性的方法，UNIX 被丹尼斯·里奇用编程语言 C（内核和 I/O 例外）重新编写。高级语言编写的操作系统具有的可用性，允许移植到不同的计算机平台更容易。

UNIX 在学术机构和大型企业中得到了广泛的应用，当时的 UNIX 拥有者 AT&T 公司以低廉甚至免费的许可将 UNIX 源码授权给学术机构做研究或教学之用，许多机构在此源码基础上加以扩充和改进，形成了所谓的“UNIX 变种”，这些变种反过来也促进了 UNIX 的发展，其中最著名的变种之一是由加利福尼亚大学伯克利分校开发的伯克利软件包（BSD）产品。

后来 AT&T 意识到了 UNIX 的商业价值，不再将 UNIX 源码授权给学术机构，并对之前的 UNIX 及其变种声明了版权权利。而 BSD 在 UNIX 的历史发展中具有相当大的影响力，被很多商业厂家采用，成为很多商用 UNIX 的基础。由于版权问题，4.4 BSD 完全删除了来自于 AT&T 的代码。尽管后来非商业版的 UNIX 系统又经过了很多演变，但其中有不少最终都创建在 BSD 版本上（Linux、MINIX 等系统除外）。所以从这个角度上，4.4 BSD 又是所有自由版本 UNIX 的基础，它们和 System V 及 Linux 等共同构成 UNIX 操作系统这片璀璨的星空。BSD 在发展中也逐渐派生出 3 个主要的分支：FreeBSD、OpenBSD 和 NetBSD。

此后的几十年中，UNIX 仍在不断变化，其版权所有者不断变更，授权者的数量也在增加。UNIX 的版权曾经为 AT&T 所有，之后 Novell 拥有了 UNIX，再之后 Novell 又将版权出售给了圣克鲁兹作业。有很多大公司在取得了 UNIX 的授权之后，开发了自己的 UNIX 产品，比如 IBM 的 AIX、惠普公司的 HP-UX、SUN 的 Solaris 和硅谷图形公司的 IRIX。

UNIX 因为其安全可靠、高效强大的特点在服务器领域得到了广泛的应用。直到 GNU/Linux 流行开始前，UNIX 也是科学计算、大型机、超级计算机等所用操作系统的主流。即使是现在，其仍然被应用于一些对稳定性要求极高的数据中心之上。

操作系统家族简图如图 1-1 所示。

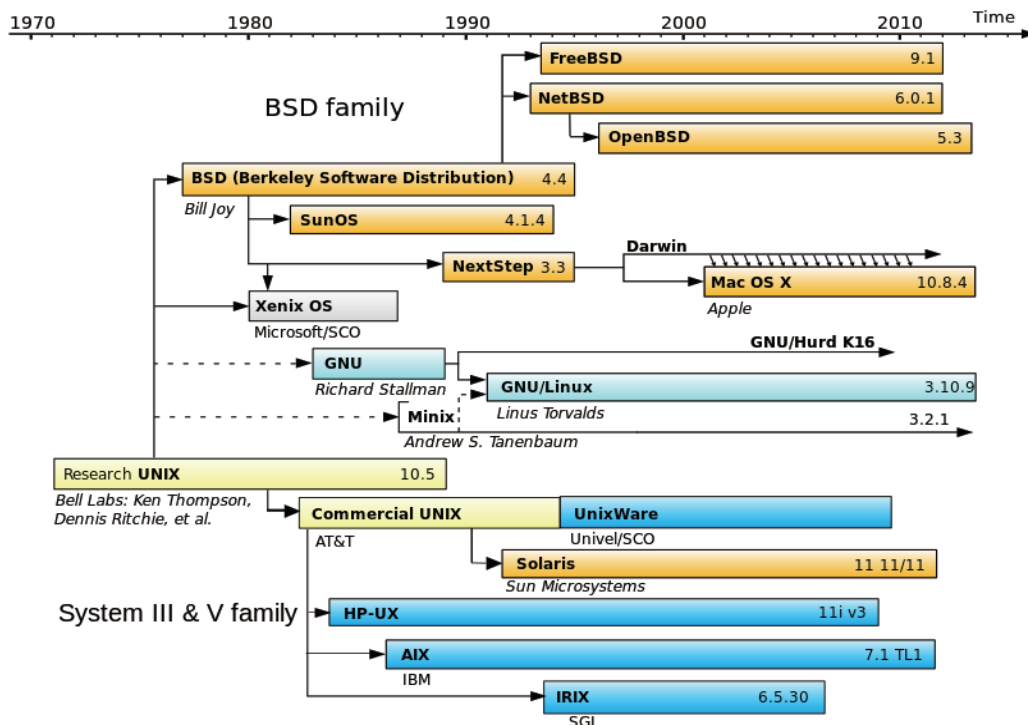


图1-1 操作系统家族

**NOTE**

值得一提的是，BSD UNIX 最先实现了 TCP/IP，除此之外，伯克利分校还开发了现代计算机领域广泛使用的 DB 和 DNS，非常的了不起。

### 3. GNU 计划

1983 年，理查德·马修·斯托曼（Richard M. Stallman）创立了 GNU 计划。这个计划有一个目标，是为了发展一个完全自由的类 UNIX 操作系统。自 1984 年发起这个计划以来，在 1985 年，理查德·马修·斯托曼发起自由软件基金会并且在 1989 年撰写了 GPL 协议（开源软件最重要的版权协议之一）。20 世纪 90 年代早期，GNU 开始大量产生或收集各种系统所必备的组件，如库、编译器、调试工具、文本编辑器、网页服务器，以及一个 UNIX 的用户界面（UNIX Shell），但是像一些底层环境，如硬件驱动、守护进程运行内核（Kernel）仍然不完整和陷于停顿。GNU 计划中是在马赫微核（Mach Microkernel）的架构之上开发系统内核，也就是所谓的 GNU Hurd。但是这个基于 Mach 的设计异常复杂，发展进度相对缓慢。林纳斯·托瓦兹曾说过，如果 GNU 内核在 1991 年时可以用，他不会自己去写一个。

GNU 计划是现代软件发展的重要力量，它倡导的开放、自由（Open source, Free software），吸引了大量的企业和个人开发者参与其中，为各个开源软件项目贡献代码，使得开源软件蓬勃发展，这也是 Linux 迅速壮大并逐渐流行的基础。

### 4. MINIX

MINIX 是一个轻量的小型类 UNIX 操作系统，是为在计算机科学用作教学而设计的，作者是安德鲁·斯图尔特·塔能鲍姆（Andrew Stuart Tanenbaum）。从第三版开始，MINIX 成为自由软件，而且被重新设计。

因为 AT&T 的政策改变，在 Version 7 UNIX 推出之后，发布了新的使用条款，将 UNIX 源代码私有化，在大学中不再能使用 UNIX 源代码。塔能鲍姆教授为了能在课堂上教授学生操作系统运作的细节，决定在不使用任何 AT&T 源代码的前提下，自行开发与 UNIX 兼容的操作系统，以避免版权上的争议。他以小型 UNIX（mini-UNIX）之意，将它称为 MINIX。

### 5. Linux 诞生

1991 年，芬兰人林纳斯·托瓦兹在赫尔辛基大学上学，对操作系统很好奇，并且对 MINIX 只在教育学术上使用的设计很不满意，于是他决定写一个更加实用的操作系统，这就是后来的 Linux 内核。

林纳斯·托瓦兹开始在 MINIX 上开发 Linux 内核，为 MINIX 写的软件也可以在 Linux 内核上使用。后来 Linux 成熟了，可以在自己上面开发自己了。为了让 Linux 可以在商业上使用，林纳斯·托瓦兹决定改变他原来的协议（这个协议会限制商业使用），使用 GNU GPL 协议来代替。采用 GPL 协议发布的 Linux 受到全世界开发者的广泛关注和参与，开发者致

力于融合 GNU 元素到 Linux 中，开发出一个有完整功能的、自由的操作系统。Linux 诞生路线如图 1-2 所示。

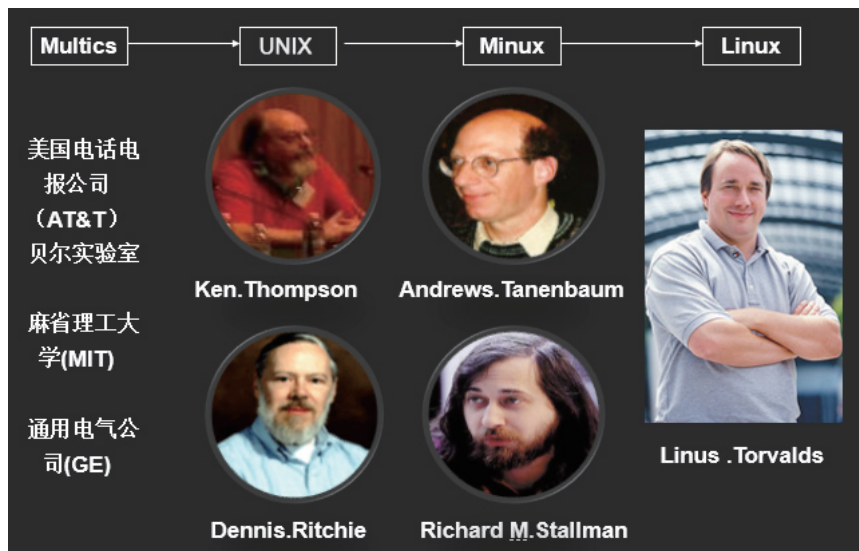


图1-2 Linux诞生

1994 年 3 月，Linux 1.0 版正式发布，Marc Ewing 成立了 Red Hat 软件公司，成为最著名的 Linux 经销商之一。

#### NOTE

1991 年 10 月，赫尔辛基大学学生林纳斯·托瓦兹在 comp.os.minix 上发布一则信息：“Hello everybody out there using minix- I'm doing a (free) operation system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.”

## 1.2 Linux操作系统的基本结构

基于 Linux 的系统是一个模块化的类 UNIX 操作系统。Linux 操作系统的大部分设计思想来源于 20 世纪 70 ~ 80 年代的 UNIX 操作系统所创建的基本设计思想。Linux 系统使用单内核，由 Linux 内核负责处理进程控制、网络，以及外围设备和文件系统的访问。在系统运行的时候，设备驱动程序要么与内核直接集成，要么以加载模块形式添加。

Linux 具有设备独立性，它的内核具有高度适应能力，从而给系统提供了更高级的功能。GNU 用户界面组件是大多数 Linux 操作系统的重要组成部分，提供常用的 C 函数库、Shell，还有许多常见的 UNIX 实用工具，可以完成许多基本的操作系统任务。大多数 Linux 系统使





用的图形用户界面创建在 X Window 系统之上，由 X Window 系统通过软件工具及架构协议来创建操作系统所用的图形用户界面。

已安装的 Linux 操作系统包含如下一些组件。

- 启动程序，例如 GRUB 或 LILO。该程序在计算机开机启动的时候运行，并将 Linux 内核加载到内存中。
- init 程序。init 是由 Linux 内核创建的第一个进程，称为根进程，所有的系统进程都是它的子进程，即所有的进程都是通过 init 启动。init 启动的进程如系统服务和登录提示（图形或终端模式的选择）。
- 软件库包含代码。可以通过运行的进程在 Linux 系统上使用 ELF 格式来执行文件，负责管理库使用的动态链接器是 ld-linux.so。Linux 系统上最常用的软件库是 GNU C 库。
- 用户界面程序，如命令行 Shell 或窗口环境。

## 1.3 Linux应用方向

经过许多年的发展，Linux 已成为最流行的操作系统之一，广泛应用于教育、科研、军事、企业及个人计算机领域。因为良好的移植性、硬件兼容性、稳定高效，使它可以方便并可靠地部署在超级计算机、工作站、数据存储、网络服务器、嵌入式设备之上。

Linux 系统典型的应用包括如下几个方面。

- 超级计算机。在 TOP 500 中，有 485 台运行 Linux 系统，超过 97%（2014 年 6 月）。
- 服务器。Linux 发行版一直被用来作为服务器的操作系统，并且已经在该领域中占据重要地位。Linux 发行版是构成 LAMP（Linux 操作系统、Apache、MySQL、Perl / PHP / Python）的重要部分。LAMP 是一个常见的网站托管平台，在开发者中已经得到普及。
- 工作站。《泰坦尼克号》、《我是传奇》、《指环王》、《星球大战》、《哈利波特》、《怪物史莱克》、《2012》、《阿凡达》等特效制作依赖于 Linux 的集群系统完成。
- 个人计算机。随着 X Window 的加入，以及桌面环境发展和应用软件的极大丰富，Linux 在图形界面易用性上也取得了长足的进步，产生了诸如 Ubuntu、Fedora 等优秀的桌面系统。
- 嵌入式设备。Linux 的低成本、强大的定制功能及良好的移植性能，使得 Linux 在嵌入式系统方面也得到广泛应用。比如数字视频系统、音频系统、车载系统、光源系统、智能家居系统采用了定制的 Linux；在网络防火墙和路由器上也大多使用了定制的 Linux。
- 在智能手机、平板电脑等移动设备方面，基于 Linux 内核的操作系统也成为最广泛的



操作系统。比如 Android、Sailfish、Firefox OS、Ubuntu Touch 等。

- 云计算。全球最大的云计算服务商 Amazon EC2 云完全构建于 Linux 架构之上；流行的 Openstack 云计算解决方案基于 Linux 系统部署。

## 1.4 Linux的特性

Linux 具有以下特性。

- Linux 是一种 UNIX-Like 操作系统，它遵循 POSIX 标准，运行在 UNIX 下的软件很容易移植到 Linux 下，这使得 Linux 立刻拥有了大量优秀的软件。同时，Linux 与 UNIX 非常相似，而它的开发人员大都拥有 UNIX 的背景。
- 使用 Linux，包括对它的复制、修改、再发布，在遵循 GPL 的协议下，你不会有版权问题的担心，对于企业部署可以极大地降低成本。而正因为支持 Linux 平台不会依赖于任何一家私有软件公司，所以各大软硬件厂商都支持并发展 Linux，如 Red Hat、IBM、Intel、Dell、Oracle、VMware、Google 等。
- 由于 Linux 的开发是基于 Internet 由社区开发的，并有众多的支持者进行测试和 bug 提交，所以使得 Linux 拥有更快的更新速度、更透明的漏洞修补和功能迭进。
- Linux 继承了 UNIX 多用户多任务的设计理念，允许多人同时上线工作，并合理分配资源。严格的用户权限管理使得不同的使用者之间保持高度的保密性和安全性。
- Linux 系统使用相对较少的硬件资源，你甚至可以找出一台你已经丢弃的古董计算机，在上面安装 Linux，在上面跑一些网络服务，你会惊讶地发现它竟然非常流畅，一般情况下，你不用担心它会越来越慢。
- Linux 独特的内核设计决定了它的网络性能极为出色，不少的网络设备厂商直接基于 Linux 开发网络路由、防火墙设备。
- 正如前面提到，Linux 得到来自各大软硬件厂商的支持，特别是企业级应用。
- Linux 的资源丰富，本身的工具和软件已经自带了详细的使用文档和大量的帮助信息。而且互联网上也有众多乐于分享和帮助的 Linux Fans，如果遇到问题，你甚至可以直接咨询软件的开发者。

## 1.5 Linux的发行版

Linux 发行版指的就是通常所说的“Linux 操作系统”，它可能是由一个组织、公司或者个人发布的。Linux 主要作为 Linux 发布版（通常被称为 distro）的一部分而使用。通常来讲，



一个 Linux 发布版包括 Linux 内核、将整个软件安装到计算机上的一套安装工具、各种 GNU 软件，以及其他一些自由软件，在一些特定的 Linux 发布版中也有一些专有软件。发行版为许多不同的目的而制作，包括对不同计算机硬件结构的支持、对一个具体区域或语言的本地化、实时应用和嵌入式系统。目前，数百个 Linux 发行版被积极地开发，被广泛应用的发行版有如下几个。

### 1. Red Hat Enterprise Linux

RHEL 是 Red Hat (Red Hat) 公司的企业版 Linux 系统，因其稳定强大、各大厂商认证和良好的技术支持，其在 Linux 服务器上市场占有率超过 50% 的份额。采用 RPM 的包管理方式，很多发行版都或多或少地受到它的影响。

### 2. CentOS

CentOS 由社区开发并维护，基于 RHEL，并与 RHEL 版本号保持一致，致力于提供一个自由使用且稳定的 RHEL。开发者直接修改 RHEL 的源代码，去除了 Red Hat 的商标和商业服务组件，修复了很多存在的 bug。其拥有自己的软件仓库，提供免费的在线更新程序。

### 3. Fedora

Fedora 主要由 Red Hat 主持的社区 Linux 项目，采用同样的 RPM 包管理，致力于最新技术的开发和引入。经过测试稳定且有价值的技术将被 RHEL 吸纳。坚持每半年发布一个版本。

### 4. SUSE Linux

SUSE Linux 是在欧洲非常流行的 Linux 发行版，以界面华丽和简单易用著称，2004 年被 Novell 收购。Novell 提供企业级的 SUSE Linux Enterprise Server | Desktop 软件和商业技术支持服务，市场占有率较高。OpenSUSE 是基于企业版的社区提供的免费 SUSE Linux。

### 5. Debian

Debian 是一个完全开放、强烈信奉自由软件的系统，由 Debian 计划组织维护，其背后没有任何营利组织的支持，开发人员完全来自于全世界各地的志愿者。Debian 基于 DEB 的包管理方式，APT 的在线软件安装更新非常方便且快速。提供超过 18 000 个软件包的支持，受到研究机构开发人员的极大欢迎。

### 6. Ubuntu

基于 Debian 开发，采用相同的 DEB 和 APT。通过精挑细选，保证软件质量，致力于开发一个简单易用的 Linux 系统。由 Canonical 支持，坚持每 6 个月发布一个版本，分别提供 6 个月和 3 年 (LTS) 的技术支持。由于其易用性和遍布世界各地的镜像源服务器，使得 Ubuntu 近年来变得非常流行。





此外还有一些非常有特色且流行的 Linux 发行版。

- Arch Linux : 一个基于 KISS (Keep It Simple and Stupid) 的滚动更新的操作系统。
- Gentoo : 一个面向高级用户的发行版, 所有软件的源代码需要自行编译。
- Elementary OS : 基于 Ubuntu, 界面酷似 Mac OS X。
- Linux Mint : 一个从 Ubuntu 派生并与 Ubuntu 兼容的系统。

如图 1-3 所示是流行的 Linux 操作系统的图标。



图1-3 流行的Linux操作系统

## 1.6 Linux的版本

Linux 系统设计对软硬件的版本要求极高, 不同的硬件平台、内核版本、系统版本、软件版本一般不能通用。对于 Linux 版本的理解, 主要是了解以下两个方面。

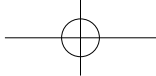
### 1. Kernel 的版本

Kernel 版本即 Linux 内核版本。Linux 内核有以下不同的命名方案。

(1) 早期版本: 第一个版本的内核是 0.01, 其次是 0.02, 0.03, 0.10, 0.11, 0.12 (第一个 GPL 版本), 0.95, 0.96, 0.97, 0.98, 0.99 及 1.0。

(2) 旧计划 (1.0 和 2.6 版之间), 版本的格式为 A.B.C, 比如 2.6.18。其中:

- A 是指大幅度转变的内核。它很少发生变化, 只有当的代码和核心发生重大变化时才会改变。在历史上曾改变两次的内核: 1994 年的 1.0 及 1996 年的 2.0。
- B 是指一些重大修改的内核。在 2.x 时代, 如果 x 为奇数指开发版本, 偶数指稳定版本。
- C 是指轻微修订的内核。这个数字当有安全补丁、bug 修复、新的功能或驱动程序时, 便会发生变化。



(3) 自 2.6.0 (2003 年 12 月) 发布后, 人们认识到, 更短的发布周期将是有益的。自那时起, 版本的格式为 A.B.C.D, 其中:

- A 和 B 是无关紧要的。
- C 是内核的版本。
- D 是安全补丁。

(4) 自 3.0 (2011 年 7 月) 发布后, 版本的格式为 3.A.B, 其中:

- A 是内核的版本。
- B 是安全补丁。

我们可以到 [www.kernel.org](http://www.kernel.org) 查看或下载最新的内核版本。

## 2. 操作系统的版本

我们在生产和学习中使用的一般是完整 Linux 操作系统发行版。每个不同的发行版都有自己的版本号命名规则。以 Red Hat 的 Linux 企业版 (Red Hat Enterprise Linux) 为例:

目前最新的版本是 6.x 和 5.x。其中 6 (或 5) 表示主版本号, 大版本的更新每三年一次; x 表示次版本号, 次版本的更新每 6 个月一次。

Red Hat Enterprise Linux 系统映象的文件名如下:

- rhel-server-6.5-i386-dvd.iso。
- rhel-server-6.5-x86\_64-dvd.iso。
- rhel-server-5.10-i386-dvd.iso。
- rhel-server-5.10-i386\_64-dvd.iso。

其中 i386 表示支持 32 位 CPU 平台, x86\_64 表示支持 64 位 CPU 平台。



### NOTE

在本书编写的时间点上, 最新版的 RHEL 7 企业级操作系统已经发布。



### REAL-WORK TIP

在生产环境中, 出于对系统性能和功能的要求, 多采用 64 位的系统。

## 1.7 其他

### 1. 开源软件和开源软件运动

开放源代码软件就是在 GNU 通用公共许可证 (GPL) 或其他开源软件协议下发布的软件, 以保障软件用户自由使用及接触源代码的权利。这同时也保障了用户自行修改、复制及再分



发的权利。简而言之，所有公布软件源代码的程序都可以称为开放源代码软件。开放源代码有时不仅仅指开放软件的源代码，它同时也是一种软件开发模式的名称。使用开放源代码开发模式的软件代表就有 Linux 操作系统。

开放源代码软件运动是一个主要由程序工程师及其他计算机用户参与的声势浩大的运动。它是自由软件运动的一个分支，但二者的差别并不明显。一般而言，自由软件运动是基于政治及哲学思想（有时被称为所谓黑客文化）的理想主义运动，而开放源代码运动则主要注重程序本身的质量提升。

## 2. Linux 的标志

Linux 的标志和吉祥物是一只名字叫作 Tux 的企鹅，如图 1-4 所示。标志的由来是因为 Linus 在澳大利亚时曾被一只动物园里的企鹅咬了一口，印象深刻，便选择了企鹅作为 Linux 的标志。更容易被接受的引申的说法是：企鹅代表南极，而南极又是全世界所共有的一块陆地，这也就代表 Linux 是所有人的 Linux。

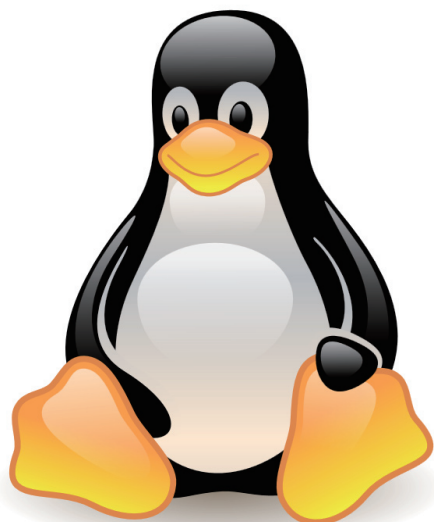


图1-4 Linux标志

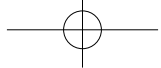


## 本章总结

Linux 是一种 UNIX-Like 操作系统。严格来说，Linux 只是一个操作系统内核。大多数 Linux 发行版是由操作系统内核加上 GNU 的软件或工具形成完整的操作系统。

### 章节重点：

- 了解 Linux 的行业应用，特别要注意在企业的核心技术应用。



- 了解 Linux 的基本结构和技术特点，理解 Linux 为什么会被广泛应用到生产生活的各个领域。
- 了解什么是开源软件，以及开源软件的优缺点。
- 了解 Linux 各大主流发行版各自的特点。



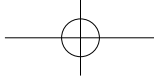
## 基础技能演练

1. 上网查找并阅读关于 UNIX、GNU、MINIX、Linux、开源软件、自由软件、GPL 及其他开源软件协议的相关资料，加深对 Linux 系统及开源软件的理解。
2. 访问本课程中提到的 Linux 发行版的官方网站，找到相关介绍和桌面截图信息。
3. 讨论和思考学习 Linux 相关知识的意义，以及如何快速、高效地学习。



## 提高技能实践

1. 通过查阅资料和结合自己的使用经验，阐述 Linux 系统和 Windows 系统有何不同。
2. 查阅资料，了解具备 Linux 相关技能的技术人员在企业的需求现状。



## 第2章

# 安装Linux

安装 Linux 系统是一个简单的过程。但是在服务器或测试工作站上顺利地完整安装一个合理的应用系统，则需要了解硬件规划、软件选择、安装方式、过程控制等方方面面，甚至一些安装过程中的排错技巧。本章主要介绍在 Intel 和 AMD 32 位有 64 位平台上的 Red Hat 企业版 Linux 的安装和引导。



### NOTE

对于其他平台（如 IBM POWER）Linux 系统的安装，已兴起并日益广泛应用的云架构平台、虚拟化平台下 Linux 系统的安装部署等知识技巧将在本系列教程高级部分中涉及。

## 2.1 获取Red Hat企业版Linux

如果您有 Red Hat 订阅，您可以在 Red Hat 客户门户网站的软件 & 下载中心下载 Red Hat 企业版 Linux 6（Red Hat Enterprise Linux 6 或简称 RHEL 6）安装 DVD 的 ISO 映象文件。注意：您只需要选择该产品的最新版本。每个发行版本都是操作系统的完整功能版本，因此不需要之前的发行版本。如果您要在服务器中安装 Red Hat 企业版 Linux，请确定您选择 server 发行版；如果您要在客户端机器中安装，则请选择 desktop 发行版。您也可以选择 32 位或 64 位版本。Red Hat 企业版 Linux 的每个版本都是单一 DVD 的 ISO 映象文件，大小为 3 ~ 4GB。

如图 2-1 所示，同一页中还有 Red Hat 企业版 Linux 源代码 ISO 映象文件链接。您不需要下载这个源代码就可安装 Red Hat 企业版 Linux 操作系统。

该页还包含 boot.iso 的链接，单击这个链接下载名为 rhel-variant-version-architectureboot.iso 的 ISO 映象文件，它的大小约为 200MB。您可以使用这个文件生成最小引导介质（如可



引导 CD、DVD 或者 USB 设备)，您可在计划使用硬盘或者网络连接中的安装源完成安装时使用这些设备引导系统。

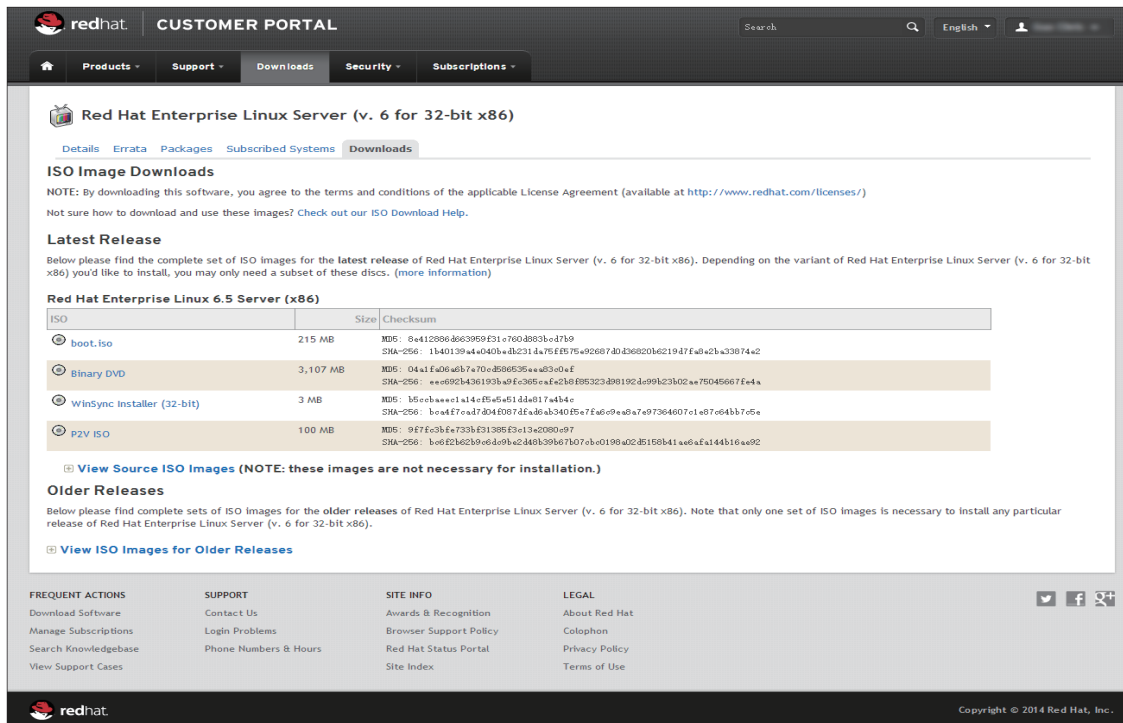


图2-1 RHEL ISO官方下载页面

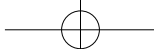
正如图 2-1 所示的内容中显示，目前 Red Hat 企业版 Linux 系统最新版信息如下：

```
Red Hat Enterprise Linux 6.5 Server (x86)
Binary DVD
3,107 MB
MD5: 04a1fa06a6b7e70cd586535eea83c0ef
SHA-256: eec692b436193ba9fc365cafe2b8f85323d98192dc99b23b02ae75045667fe4a
boot.iso
215 MB
MD5: 8e412886d663959f31c760d883bcd7b9
SHA-256: 1b40139a4e040bedb231da75ff575e92687d0d36820b6219d7fa8e2ba33874e2
```



## NOTE

到每个映像文件的链接都有 MD5 和 SHA-256 校验码。在您的下载完成后，使用校验码工具如 md5sum 或 sha256sum 来生成本地文件拷贝的检验码。如果您生成的值和网站上的值相匹配，那说明这个映像文件与官方提供的一致，是真实的且未被破坏的。

**REAL-WORK TIP**

Red Hat 企业版 Linux 是以稳定著称的优秀的开源商业软件发行版，推荐企业用户购买官方订阅应用于生产环境。在学习测试环境或重要性级别低的服务器上，可以使用 Red Hat 企业版 Linux 的社区衍生版 CentOS 替代，该发行版几乎 100% 兼容 Red Hat 企业版，可免费使用并提供在线更新。

**创建安装介质**

您可以使用您的计算机系统上的刻录软件（如 Nero Burning ROM）将下载的 ISO 映像文件刻录生成可引导安装 DVD。如果您是将 Linux 系统安装到虚拟机，则不用刻录成 DVD，直接连接 ISO 映像文件到虚拟光驱，虚拟机选择从光驱启动即可（参见附录 A）。

表 2-1 给出了可用于不同架构的引导和安装介质类型并标注了您需要用来生成介质的映像文件。

**表2-1 制作安装介质**

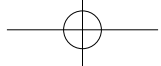
体系结构	安装 DVD	引导 CD/DVD	引导 USB 闪存驱动器
基于 BIOS 的 32 位 x86	x86 DVD ISO 映像文件	rhel-variant-version-i386-boot.iso	rhel-variant-version-i386-boot.iso
基于 UEFI 的 32 位 x86	不可用		
基于 BIOS 的 AMD64 和 Intel 64	x86_64 DVD ISO 映像文件（安装 64 位操作系统）或者 x86 DVD ISO 映像文件（安装 32 位操作系统）	rhel-variant-version-i386_64-boot.iso 或 rhel-variant-version-i386-boot.iso	rhel-variant-version-i386_64-boot.iso 或 rhel-variant-version-i386-boot.iso
基于 UEFI 的 AMD64 和 Intel64	x86_64 DVD ISO 映像文件	rhel-variant-version-i386_64-boot.iso	efidisk.img（使用 x86_64 DVD ISO 映像文件）
POWER（仅限 64 位）	ppc DVD ISO 映像文件	rhel-variant-version-ppc64-boot.is	不可用
System Z	s390 DVD ISO 映像文件	不可用	不可用

其中 variant 是 Red Hat 企业版 Linux 的变体（如 server 或 desktop），version 是最新的版本号（如 6.5）

## 2.2 安装前的准备

### 1. 选择升级还是安装

通常 Red Hat 不支持 Red Hat 企业版 Linux 主要版本间的平滑升级，例如，RHEL 5 和 RHEL 6 都是 Red Hat 企业版 Linux 的主要版本。Red Hat 强烈推荐您在一个主要版本升级到另一个主要版本时进行全新安装。



## 2. 硬件条件

RHEL 6 能够兼容最近两年内出品的大多数硬件，如果您需要了解最新的硬件支持列表，可以访问 <https://hardware.redhat.com/> 查询。另外，运维技术人员应该根据实际的业务需求，选择配置最合理优化的硬件平台。



### REAL-WORK TIP

虽然大多数情况下我们无法规划硬件规格与配置，但还是有必要了解一下当前的硬件是否满足（或过剩）于业务应用。比如一个文件服务器可能需要较多的存储空间，并做好磁盘阵列。

## 3. 软件选择

对于软件的选择，虽然在安装之前就要进行规划，但如何规划却要求对 Red Hat Linux 系统非常了解。比如选择较老但是稳定的 RHEL 5 还是最新的 RHEL 6（本教材默认）；是选择 32 位还是 64 位的系统；如何规划分区及选择安装哪些软件包。



### REAL-WORK TIP

因为目前硬件的发展及软件功能上的差异，在企业级应用环境推荐安装 64 位系统。



### REAL-WORK TIP

当对 Linux 使用非常熟练后，有经验的工程师通常在服务器上仅安装最小化的 Linux 系统，然后再按需添加需要的功能。

## 4. 备份数据

安装前必须备份原来系统里重要的数据。

## 5. 安装方式

RHEL 6 支持多种安装方式：

- 从 CD-ROM 引导安装。
- 从硬盘驱动器引导安装。
- 从 USB 闪存驱动器引导安装。
- 从 PXE 网络引导安装。

将安装映象文件置于 DVD 文件、本地硬盘、移动存储设备或网络文件服务器上。网络传输协议支持 FTP、HTTP、HTTPS、NFS。





## 2.3 开始安装

### 1. 启动安装程序

要使用 Red Hat 企业版 Linux DVD 或者最小引导介质启动安装程序，请按照以下步骤执行：

- (1) 断开所有与您安装不必要的外部固件或者 USB 磁盘的连接。
- (2) 打开您的计算机系统。
- (3) 在计算机中插入该介质。
- (4) 关闭计算机并将引导介质留在里面。
- (5) 打开您的计算机系统。

您需要按具体的按键或者组合键从该介质引导。在大多数计算机中会在开机后很短的时间内在屏幕中出现一个信息，通常类似“Press F12 to select boot device”，但不同的计算机中的具体文字及您要按的按键有很大不同。查看您计算机或主板的文档，或者从硬件生产商或零售商处寻求支持。

如果您的计算机不允许您在启动时选择引导设备，您可能需要将您系统的基本输入/输出系统（BIOS）配置为使用该介质引导。



#### NOTE

UEFI 和 BIOS 的引导配置有很大差别，因此安装的系统必须使用安装时所用的同一固件引导。您不能在使用 BIOS 的系统中安装操作系统后，再在使用 UEFI 的系统中引导这个安装。

### 2. 引导菜单

引导介质显示有几个选项的图形引导菜单。如果在 60 秒内没有按任何键，则运行默认引导选项。要选择默认，您可以等待计数器超时或者按 Enter 键。要选择默认选项外的不同选项，请使用键盘中的箭头键并在选中正确选项时按 Enter 键。引导菜单如图 2-2 所示。

引导菜单选项如下。

- 安装或者升级现有系统。这个选项是默认的。选择这个选项将在您的计算机系统中使用图形安装程序安装 RHEL 6。
- 使用基本视频驱动程序安装系统。这个选项允许您在安装程序无法为您的显卡载入正确驱动程序时使用图形模式安装 RHEL 6。如果您在使用「安装或者升级现有系统」时乱屏或者成空白屏幕，重启您的计算机并尝试使用这个选项。



图2-2 启动引导界面

- 救援安装的系统。选择这个选项修复已安装但无法正常引导的 Red Hat 企业版 Linux 中的问题。虽然 Red Hat 企业版 Linux 是非常稳定的计算机平台，但偶尔也会因为人为配置错误或其他原因而发生无法引导的问题。救援环境包含可让您修复大量此类问题的工具程序。
- 使用本地驱动器引导。这个选项使用第一个系统的磁盘引导。如果您误用这个介质引导，请使用这个选项立刻从硬盘引导且不用启动安装程序。

**REAL-WORK TIP**

在实际安装过程中，在引导界面按 Tab 键可以编辑所选的选项；按 Esc 键进入自定义引导选项的“boot:”提示符。自定义引导选项如 linux text 等，将在本系列教程高级部分涉及。

## 2.4 使用anaconda安装

anaconda 是 Red Hat 企业版 Linux 和 Fedora 的系统安装管理程序。它提供图形化的安装界面和文本模式，功能包括简洁但功能强大的系统安装向导、支持配置自动安装、虚拟控制台、屏幕截图功能等。

默认情况下，anaconda 启动后进入图形安装界面。如果出现以下情况之一，安装程序将使用文本模式：



- 安装程序无法识别您计算机中的显示硬件。
- 您在引导菜单中选择文本模式安装。

以下是完整的 RHEL 6.5 的安装过程，适用于 RHEL 6.x 全系列。其他发行版有类似的安装步骤。

**Step 01** 一般情况下确定安装映象没有问题，直接选择 Skip 跳过，如图 2-3 所示。

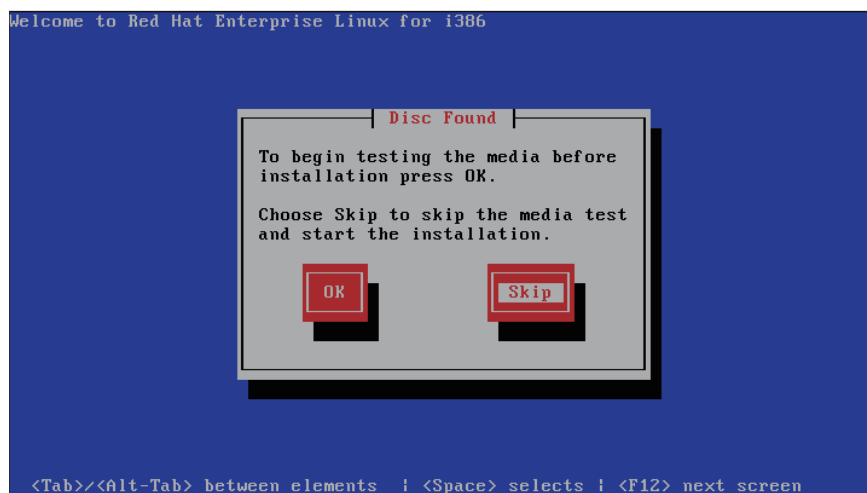


图2-3 光盘机检测

**Step 02** 单击“Next”按钮继续，如图 2-4 所示。

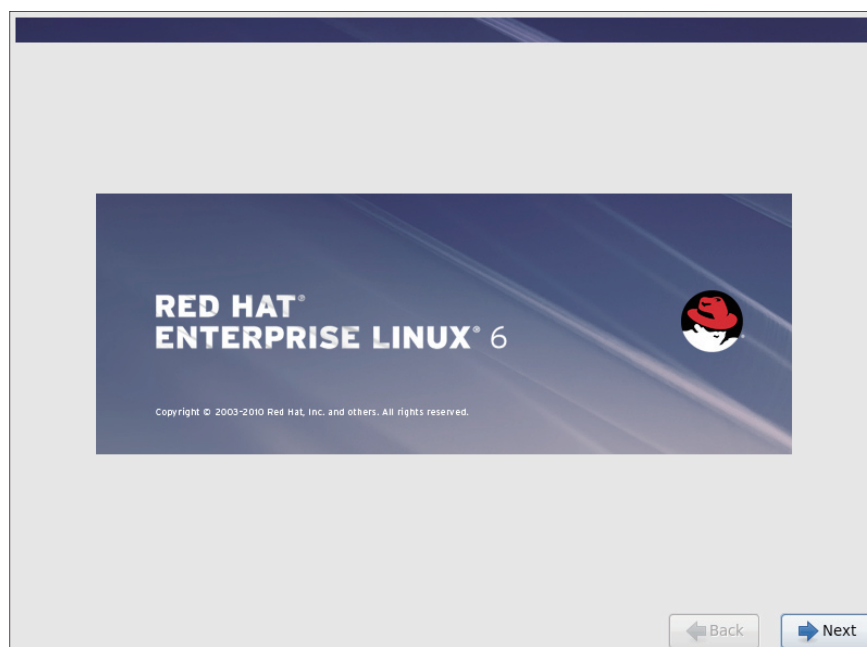


图2-4 准备开始



**Step 03** 选择安装时的首选语言，单击“Next”按钮继续，如图 2-5 所示。

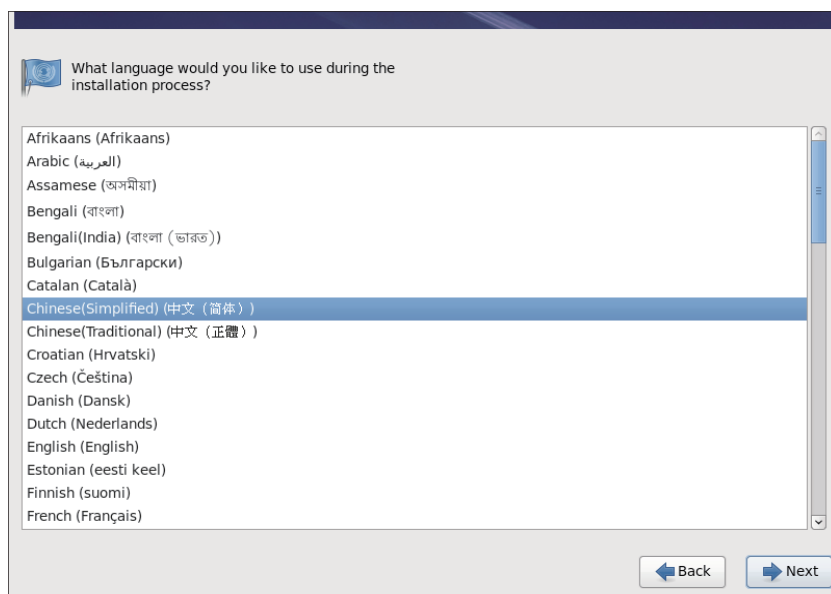


图2-5 选择安装语言

**Step 04** 选择正确的键盘布局类型，单击“下一步”按钮继续，如图 2-6 所示。

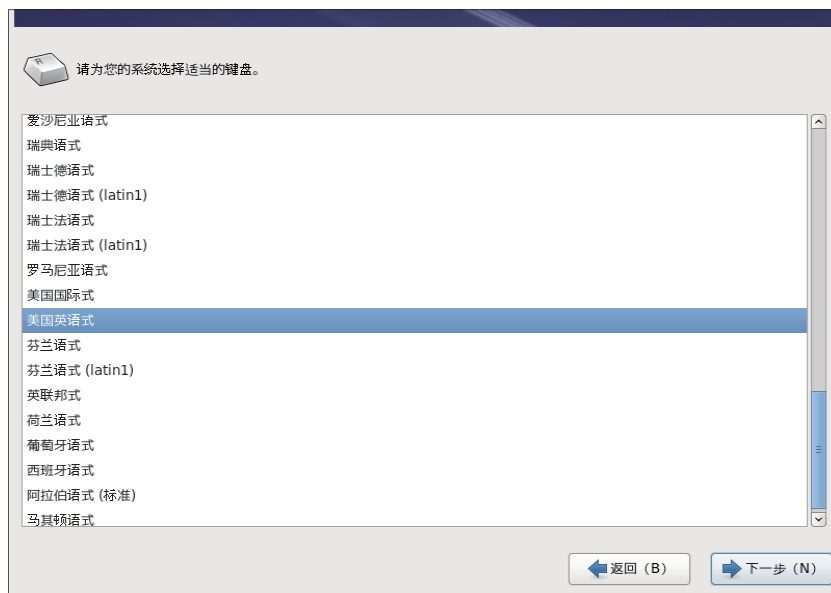


图2-6 选择键盘类型

**Step 05** 选择安装到的存储设备。大多数情况下保持默认选择即可。选择“基本存储设备”，anaconda 会自动探测附加到系统的本地存储。单击“下一步”按钮继续，如图 2-7 所示。



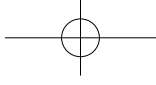
图2-7 选择存储设备

**Step 06** 发现新的虚拟磁盘，这里应单击“是，忽略所有数据”按钮。单击“下一步”按钮继续，如图 2-8 所示。



图2-8 初始化磁盘

**Step 07** 设置主机名。设置提示您提供这台计算机的主机名和域名，可以是完全限定域名（FQDN），其格式为 `hostname.domainname`；也可以是简短主机名，其格式为 `hostname`。很多网络有动态主机配置协议（DHCP）服务，它可自动提供带域名的链接系统。要允许



DHCP 服务为这台机器分配域名，请只指定简短主机名即可。单击“下一步”按钮继续，如图 2-9 所示。

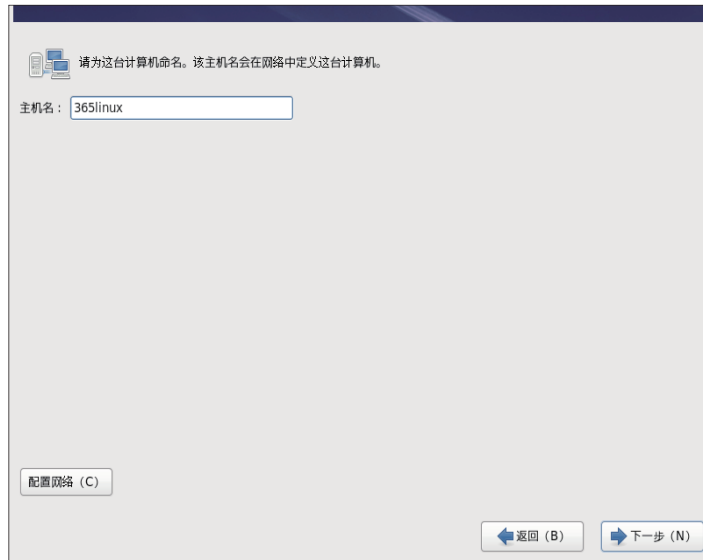


图2-9 设置主机名



**NOTE**

可以为系统起任何名字，只要完整名称是唯一的。主机名只能包含字母、数字和连字符。

**Step 08** 在图 2-9 所示的界面中单击“配置网络”按钮，弹出“网络连接”对话框，此时可使用 NetworkManager 工具配置网络连接。如图 2-10 所示，“System eth0”表示安装程序探测到的第一张有线网卡，单击“编辑”按钮即可对其进行配置。为简化安装过程，此处不进行配置。网络配置也可在系统安装完成后启动 NetworkManager 工具进行。



图2-10 设置网络连接

**REAL-WORK TIP**

如果该系统应用于生产环境，通常情况下，需要配置指定的静态 IP 地址。

**Step 09** 配置时区，如图 2-11 所示。

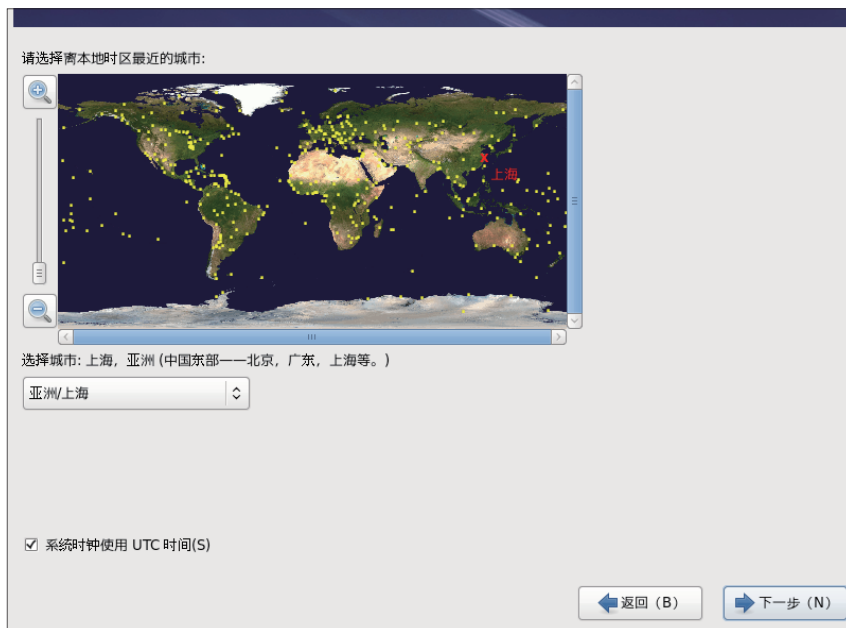


图2-11 配置时区

如果 Red Hat 企业版 Linux 是您计算机中唯一的操作系统，请勾选“系统时钟使用 UTC 时间”复选框。系统时钟是您计算机系统中硬件的一部分。Red Hat 企业版 Linux 使用时区设置来确定本地时间和系统时钟中的 UTC 之间的偏差。这个行为是使用 UNIX、Linux 和类似操作系统的标准动作。

如果您的机器还运行微软 Windows，则取消勾选“系统时钟使用 UTC 时间”复选框。微软操作系统会更改 BIOS 时钟使其与本地时间而不是 UTC 匹配，这可能导致 Red Hat 企业版 Linux 中的意外行为。

单击“下一步”按钮继续。

**Step 10** 设定 root 密码，如图 2-12 所示。

设置 root 账户和密码是安装过程中最重要的步骤之一。这个 root 账户是用来安装软件包、升级 RPM 及执行大多数系统维护工作的。作为 root 用户登录可让您完全控制您的系统。

root 密码至少有 6 个字符；您输入的密码不会在屏幕中出现。您必须输入该密码两次；如果两次不一致，安装程序会让您重新输入密码。

您应该使用您可以记住但其他人不容易猜到的字符串作为 root 密码。您的名字、电话号码、qwerty、password、root、123456 和 anteater 都是坏密码的例子。好密码应该是数字字母混合，字母包含大、小写，且不包含字典中的单词。

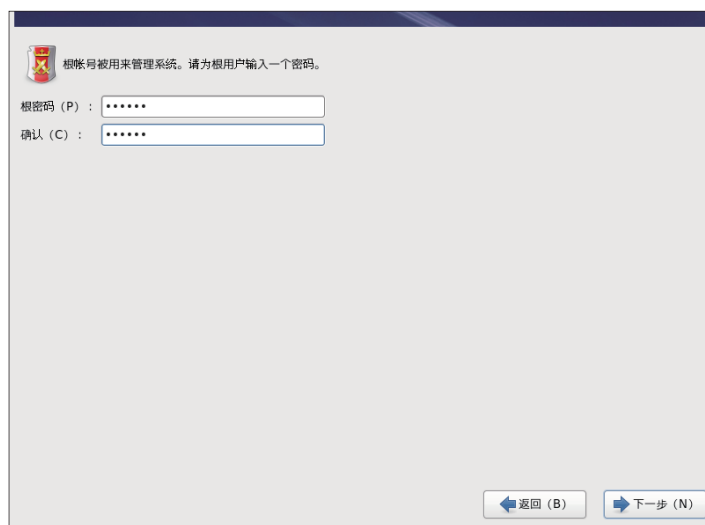
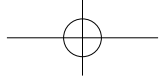


图2-12 设置root用户密码

**NOTE**

安装后 root 用户（又称超级用户）可访问、控制和修改整个系统，操作时具有一定风险。因此，为安全起见，最好只有在执行系统维护或管理时才使用 root 用户登录，而为您的日常使用创建一个非 root 的权限较小的普通账户。当在执行某些需要超级用户权限的任务时，可以使用 su 命令从普通用户切换到 root 用户。而本教程作为学习实验环境进行系统级别的设置操作，大多数以 root 用户身份执行，以此说明。

**Step 11** 如图 2-13 所示，安装系统之前的分区处理。anaconda 提供了 5 种方案可供选择。对于一个全新的系统安装，通常会选择第一种“使用所有空间”，并勾选左下角的“查看并修改分区布局”复选框来对系统默认分区进行自定义调整；或者直接选择第五种完全自定义方案。这里选择“创建自定义布局”单选按钮，单击“下一步”按钮继续。

**NOTE**

如果在现有硬盘中没有找到可读分区表，则安装程序会要求初始化该硬盘。这个操作可使该硬盘中的所有现有数据不可读。如果您的系统有一个全新硬盘且其中没有安装任何操作系统，或者您已经删除了该硬盘中的所有分区，请重新初始化驱动器。

**REAL-WORK TIP**

不同的系统管理员有自己不同的分区方案，或者项目需求决定分区方式，所以并不存在所谓标准的分区方式。但要注意两点：

- /boot 分区不能是 LVM 逻辑卷。
- 事先针对应用需求做好分区规划。



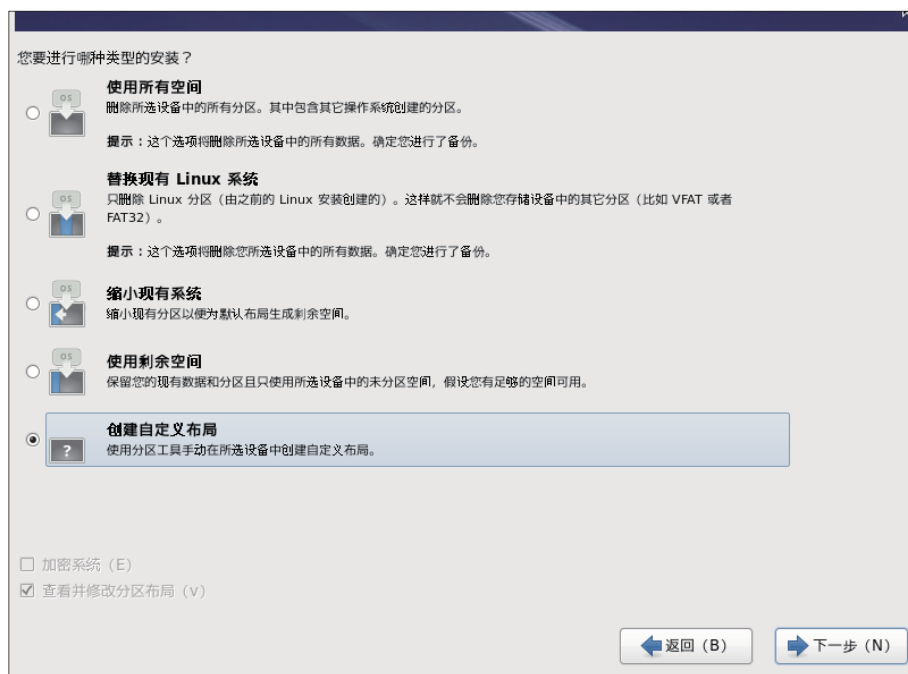


图2-13 选择分区方案

**Step 12** 如图 2-14 所示，可以看到约 20GB 是空闲的硬盘。单击“创建”按钮继续。

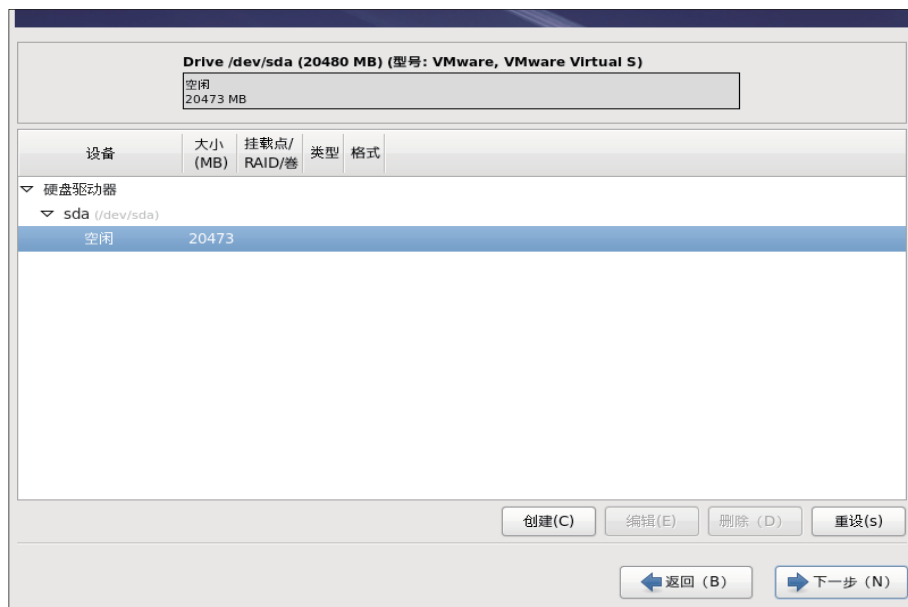


图2-14 分区规划

**Step 13** 如图 2-15 所示，选择“标准分区”单选按钮，单击“创建”按钮继续。

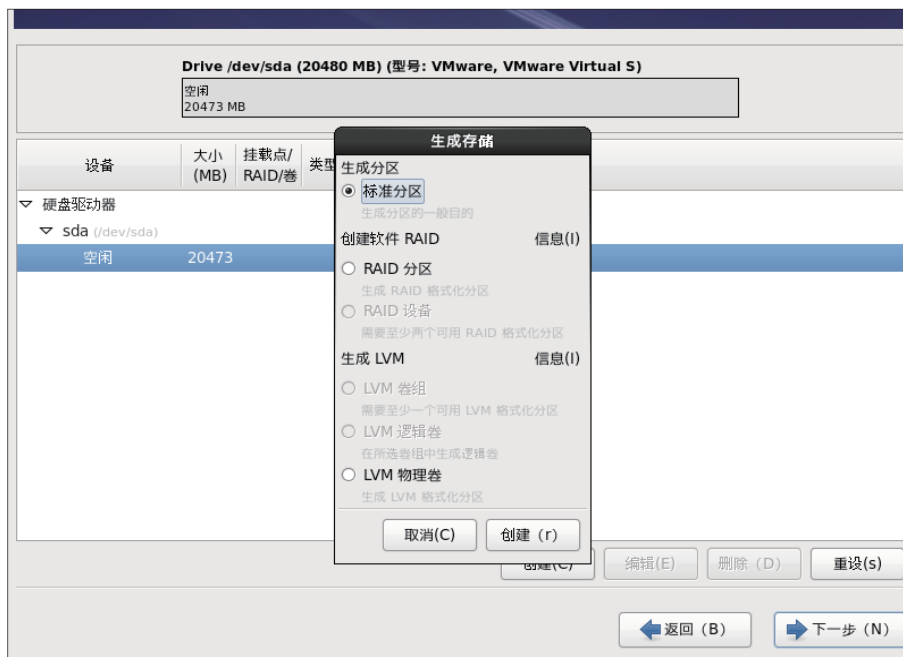


图2-15 分区过程（1）

**Step 14** 如图 2-16 所示,挂载点为“/boot”,文件系统类型为“ext4”,大小为“500MB”,单击“确定”按钮。

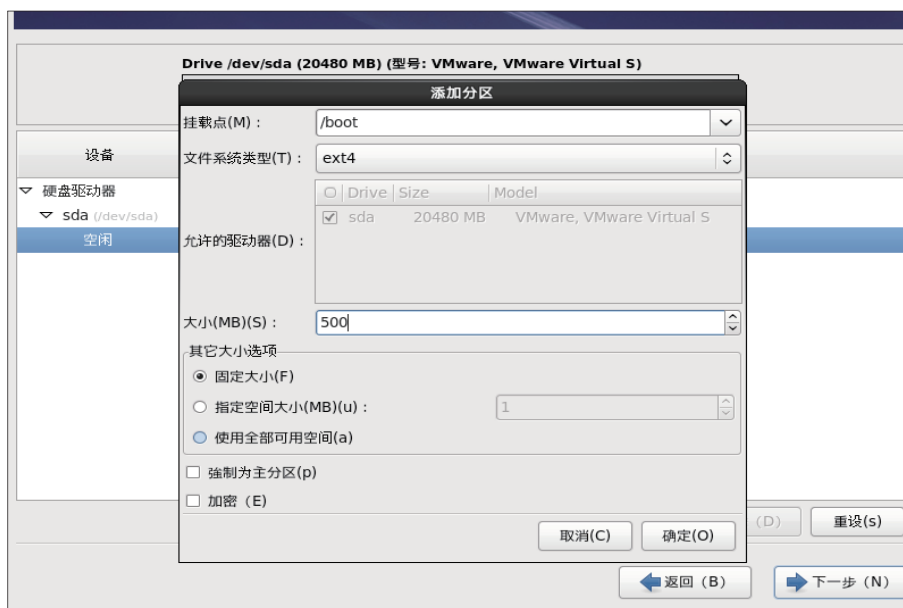


图2-16 分区过程（2）

**Step 15** 如图 2-17 所示,用同样的方法创建 swap 分区,大小为 4096MB。



图2-17 分区过程 (3)

**Step 16** 如图 2-18 所示，用同样的方法创建 / 分区（root 分区），“其他大小选项”选择“使用全部可用空间”。

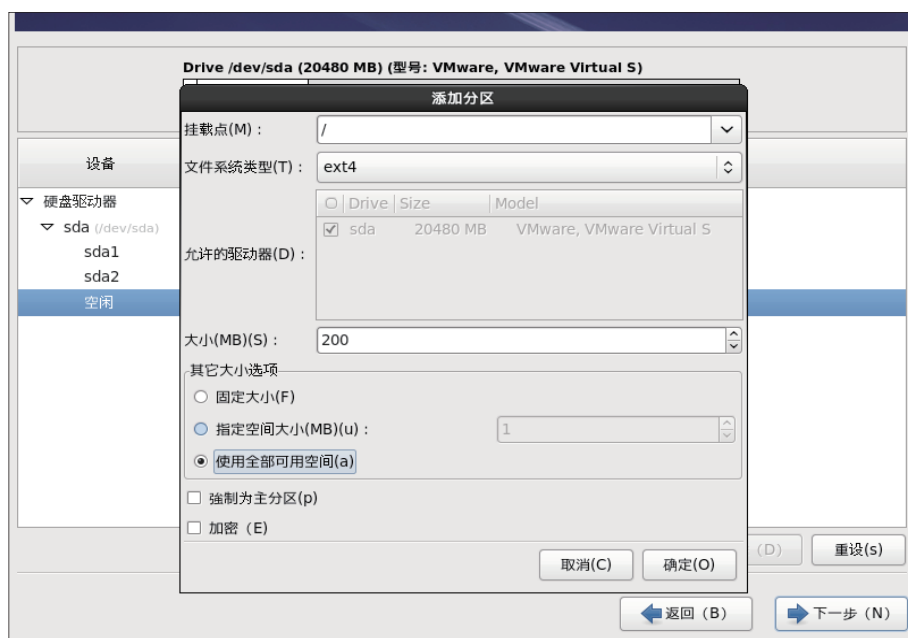
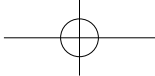


图2-18 分区过程 (4)



**Step 17** 如图 2-19 所示为最终的分区列表。单击“下一步”按钮继续。

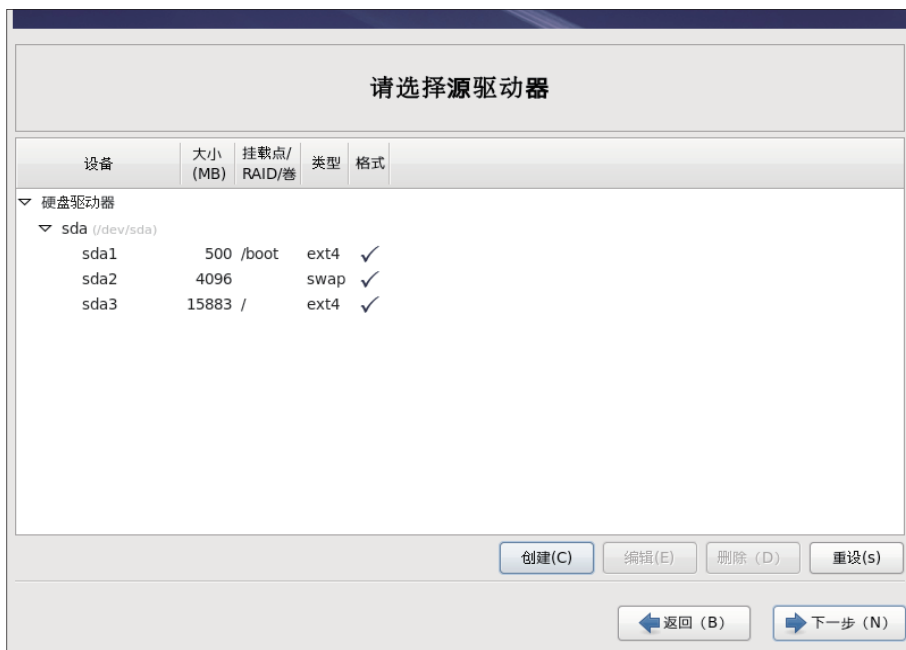


图2-19 分区完成

正如前文所述，并没有所谓的 Linux 安装标准分区方案，这里只是展示了安装时的分区过程。示例中分区表所包含的 3 个分区是大多数系统必须创建的分区，但不限于此。3 个分区的作用分别如下。

- /boot 分区。RHEL 6 最少要求 250MB 大小。挂载在 /boot 中的分区，包含操作系统内核（它可让您的系统引导 Red Hat 企业版 Linux），以及在自我引导过程中使用的文件。
- swap 分区。RHEL 最少要求 256MB 大小。swap 分区被用来支持虚拟内存。换一句话说，当内存不足以存储系统正在处理的数据时，部分数据就会被写入 swap 分区。几年前，推荐的 swap 空间的增长与系统中的 RAM 量是呈线性关系的。但由于现在系统内存大小已经增加到成百 GB，我们认识到系统需要的 swap 空间量是该系统中运行的内存负载的一个对应功能。Red Hat 官方推荐的系统 swap 空间如表 2-2 所示。

表2-2 swap官方推荐大小

系统中的 RAM 量	推荐的 swap 空间量
4 GB RAM 或者更少	最小 2 GB swap 空间
4 ~ 16 GB RAM	最小 4 GB swap 空间
16 ~ 64 GB RAM	最小 8 GB swap 空间
64 ~ 256 GB RAM	最小 16 GB swap 空间
256 ~ 512 GB RAM	最小 32 GB swap 空间



- / 分区（root 分区）。RHEL 6 最少要求 3GB 大小。它是 “/”（根目录）所在位置。在这种设置中，所有文件（除了那些保存在 /boot/ 中的文件）都位于 root 分区中。

**NOTE**

关于硬盘驱动器、分区、挂载、LVM、RAID 等细节我们将在后面的章节或本系列教程高级部分提及。

**Step 18** 如图 2-20 所示，单击“格式化”按钮继续。

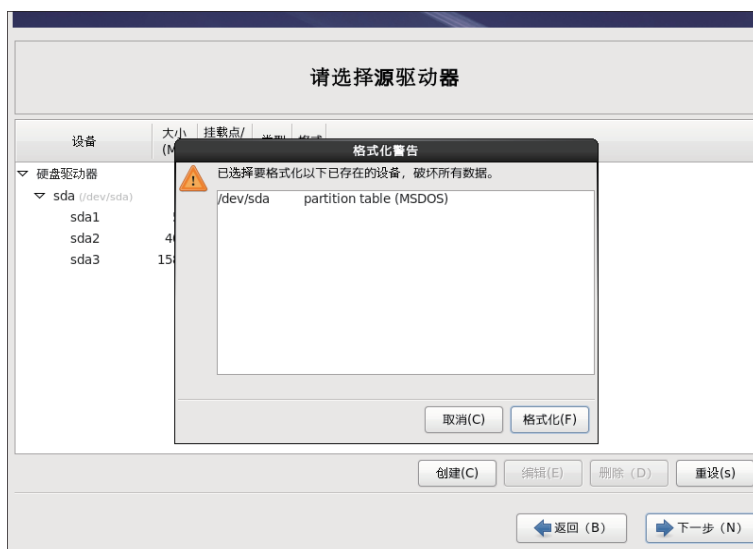
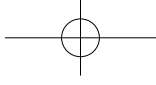


图2-20 磁盘格式化

**Step 19** 如图 2-21 所示，单击“将修改写入磁盘”按钮继续。



图2-21 写入磁盘



**Step 20** 如图 2-22 所示，等待磁盘格式化过程。

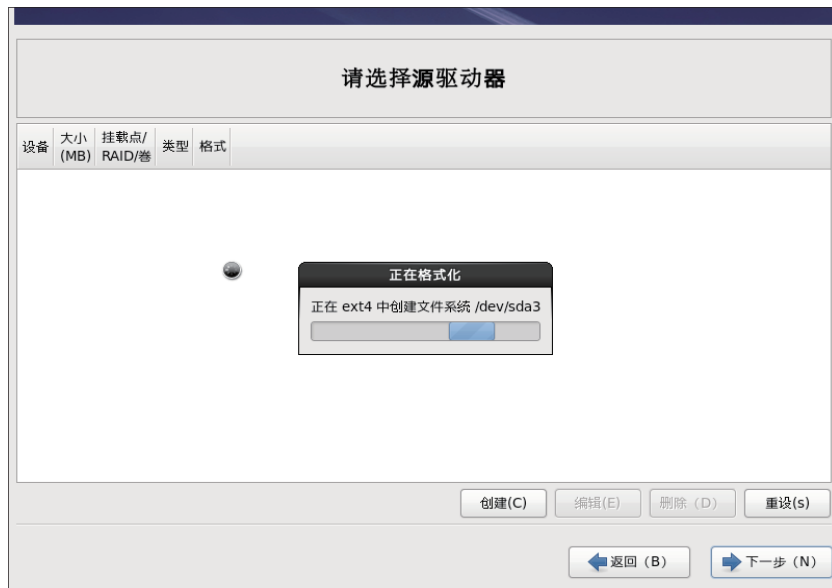


图2-22 磁盘格式化过程

**Step 21** 在硬盘上安装引导装载程序，此处使用默认设置即可，如图 2-23 所示。单击“下一步”按钮继续。

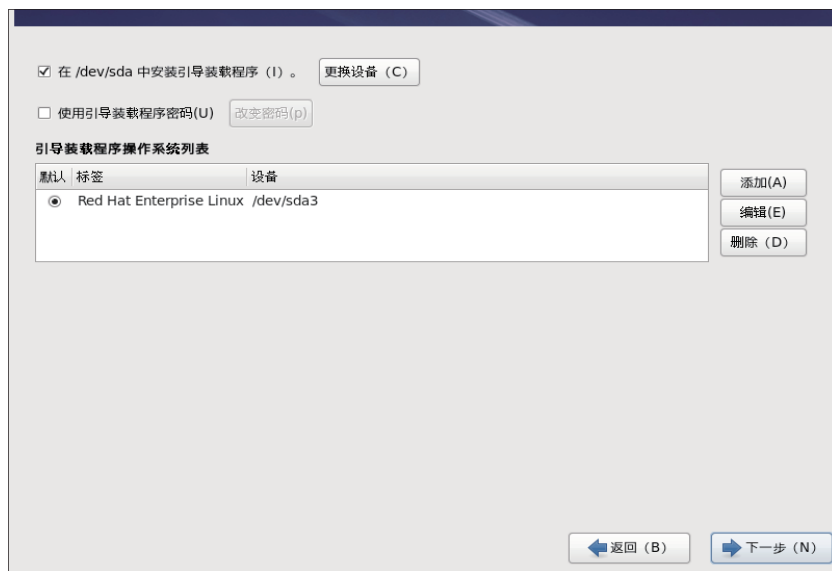


图2-23 安装引导程序

**Step 22** 软件包组的选择，如图 2-24 所示。

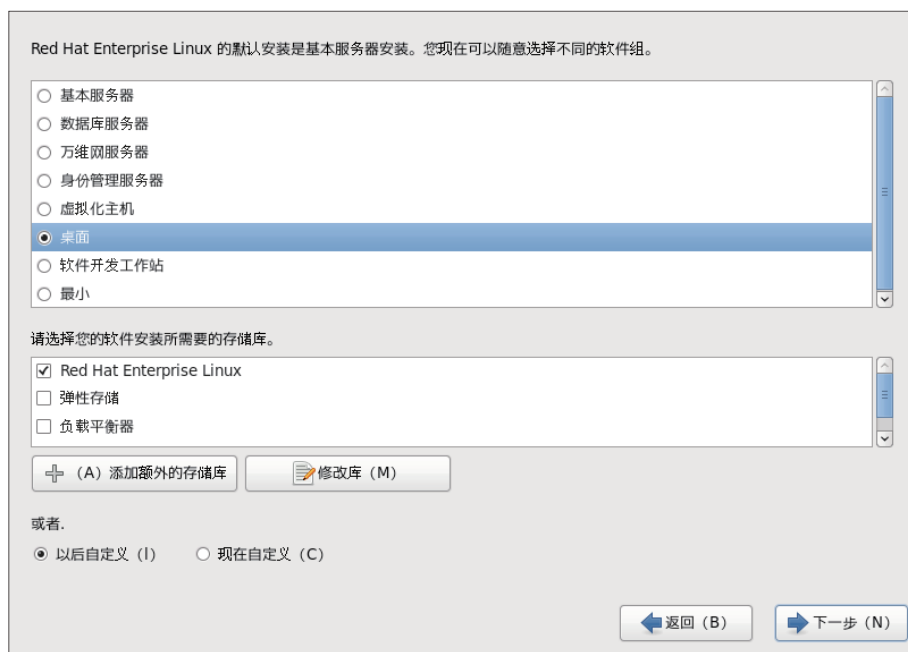


图2-24 软件包选择

默认情况下，Red Hat 企业版 Linux 安装进程载入将系统部署为基本服务器的适当软件选择。请注意这个安装不包含图形环境。要包含适合其他角色的软件选择，请选择相应的单选按钮。

- 基本服务器：这个选项提供在服务器中使用的 Red Hat 企业版 Linux 基本安装。
- 数据库服务器：这个选项提供 MySQL 和 PostgreSQL 数据库。
- 万维网服务器：这个选项提供 Apache 网页服务器。
- 身份管理服务器：这个选项提供 OpenLDAP 和 System Security Services Daemon (SSSD) 以创建标识符和验证服务器。
- 虚拟化主机：这个选项提供 KVM 和 Virtual Machine Manager 工具以创建用于虚拟机器的主机。
- 桌面：这个选项提供桌面环境和图形工具及多媒体程序。
- 软件开发工作站：这个选项提供在您的 Red Hat 企业版 Linux 中编译软件所需的工具。
- 最小：这个选项只提供运行 Red Hat 企业版 Linux 的基本软件包。最小安装为单一目的服务器或者桌面设备提供基本需要，并可在这样的安装中最大化性能和安全性。

在此处学习环境，我们选择“桌面”软件包组，单击“下一步”按钮则跳转至“安装开始”界面。如果还需要对选择的软件组件进行更细致的选择，可以选中“现在自定义”单选按钮，然后单击“下一步”按钮继续。

**Step 23** 如图 2-25 所示，安装开始。等待安装完成。

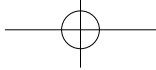


图2-25 安装开始

安装完成后，请单击“重新引导”按钮重新启动您的计算机。

用于参考，您在重启系统后可在 /root/install.log 中找到完整的安装日志。

## 2.5 安装后的配置

第一次启动 Red Hat 企业版 Linux 系统时（以图形运行级别），会出现 FirstBoot 工具，如图 2-26 所示。它可引导您完成 Red Hat 企业版 Linux 配置。使用这个工具，您可以设置系统时间和日期、安装软件、在 Red Hat 网络注册您的机器等。



图2-26 欢迎界面





## 1. FirstBoot 配置

具体操作步骤如下：

**Step 01** 单击“前进”按钮继续。

**Step 02** 在图 2-27 所示的界面中选择“是，我同意该许可协议”单选按钮，单击“前进”按钮继续。



图2-27 许可证信息

**Step 03** 这个向导是指引您注册到 Red Hat 网络来设置软件更新，需要用户购买 Red Hat 订阅。学习环境就选择“不，以后再注册”，如图 2-28 所示。单击“前进”按钮继续。再一次确认“以后再注册”，并单击“前进”按钮继续。



图2-28 设置更新



**Step 04** 创建普通用户账户。总是使用这个普通账户登录您的 Red Hat 企业版 Linux，而不使用 root 账户。填好必要的信息后，单击“前进”按钮继续，如图 2-29 所示。



图2-29 创建常用用户

**Step 05** 调整系统时钟的日期和时间。如图 2-30 所示，可以勾选“在网络上同步日期和时间”复选框将您的系统配置为使用网络时间协议（NTP）服务器以保证时钟准确。NTP 为同一网络中的计算机提供时间同步服务。互联网中有很多公共 NTP 服务器可用。



图2-30 设置时间

**Step 06** 如图 2-31 所示，选择是否在这个系统中使用 Kdump。Kdump 是内核崩溃转储机制。系统崩溃时，Kdump 会捕获系统中的信息，这对确定造成崩溃的原因至关重要。请

注意，如果您选择使用 Kdump，您将需要为其保留内存，且这个内存不能用于其他任何目的。



图2-31 配置Kdump

## 2. 配置完成

配置完成后系统需要再一次重新启动到登录界面，如图 2-32 所示。

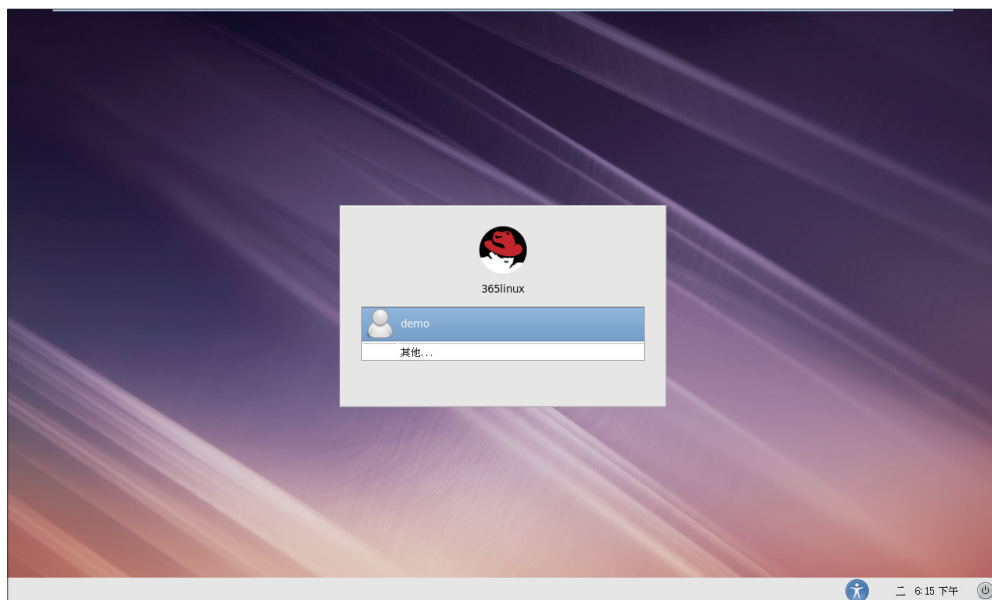
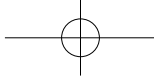


图2-32 登录界面



具体操作步骤如下：

**Step 01** 单击您的用户名，然后输入正确的密码即可登录系统。

**Step 02** 进入系统，如图 2-33 所示。至此，安装完整完成。

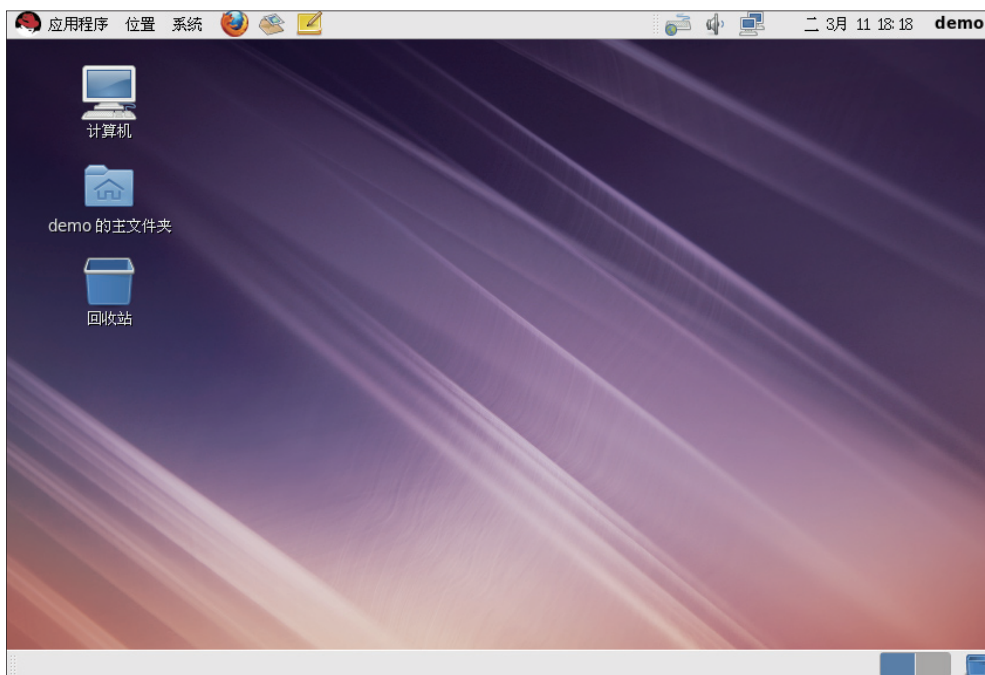


图2-33 进入系统

## 2.6 高级安装选项



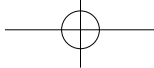
### NOTE

本节内容难度超出本课程设计，仅作为引入介绍。相关内容详解会在本系列教程高级部分中涉及。

### 1. 引导选项

Red Hat 企业版 Linux 安装系统包括一系列管理员使用的功能和选项。要使用引导选项，请在“boot:”提示符后输入 linux option。例如：

```
linux text
linux resolution=1024x768
linux vnc vncpassword=qwerty
linux ks=location/kickstart-file.cfg
```



## 2. 文本模式安装

使用 `linux text` 引导选项或安装程序无法启动图形界面时会切换到文本模式安装。文本模式呈现给您的是更简单、线形的安装过程，所以某些在图形安装模式中可以使用的选项在文本模式中无法使用，它们是：

- 配置高级存储方法，比如 LVM、RAID、FCoE、zFCP 及 iSCSI。
- 自定义分区布局。
- 自定义引导装载程序布局。
- 在安装过程中选择软件包。
- 使用 `first boot` 配置安装的系统。

## 3. 虚拟控制台

Red Hat 企业版 Linux 安装程序提供更多的安装进程对话框。您可以使用有些对话框信息，也可在 Shell 提示符后输入命令，通常用来帮助诊断安装过程中出现的问题。该安装程序在 5 个虚拟控制台中显示这些信息，您可以使用如表 2-3 所示的组合键切换到这些控制台。

表2-3 虚拟控制台

控 制 台	按 键	内 容
1	Ctrl+Alt+F1	图形化显示
2	Ctrl+Alt+F2	Shell 提示符
3	Ctrl+Alt+F3	安装日志
4	Ctrl+Alt+F4	系统相关信息
5	Ctrl+Alt+F5	其他输出信息

## 4. Kickstart 安装

创建单一的 Kickstart 配置文件，该文件记录了在安装过程中的交互式问题选项的回答。通过在安装开始阶段引入此文件来实现系统无人值守的自动安装过程。

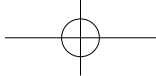
## 5. 无介质安装

通过 GRUB 或 PXE 引导安装内核和 RAM 磁盘，将安装源放置在网络服务器上，实现无介质安装 Linux 系统。通常配合 Kickstart 文件在批量部署服务器时使用。

## 6. 安装过程中的故障排除

出于调试目的，`anaconda` 将安装动作记录到 `/tmp` 目录下的文件中。包括如下几个文件。

- `/tmp/anaconda.log`：常规的 `anaconda` 信息。
- `/tmp/program.log`：`anaconda` 运行的所有外部程序。
- `/tmp/storage.log`：广泛存储的模块信息。
- `/tmp/yum.log`：YUM 软件包安装信息。
- `/tmp/syslog`：与硬件相关的系统信息。



如果安装失败，这些文件中的信息会合并到 `/tmp/anaconda-tb-identifier` 中，这里的 `identifier` 是一个随机字符串。

以上所有文件都位于安装程序的 `ramdisk` 中，因此不是恒定的。要生成永久备份，请使用安装映象中的 `scp` 将这些文件复制到网络中的另一个系统中（反之则不可）。

## 2.7 删除Red Hat企业版Linux

要从您的基于 x86 的系统中卸载 Red Hat 企业版 Linux，您必须从主引导记录（MBR）中删除 Red Hat 企业版 Linux 引导装载程序信息，同时删除包含该操作系统的所有分区。从计算机中卸载 Red Hat 企业版 Linux 主要分为以下几种情况考虑。

### 1. Red Hat 企业版 Linux 是计算机上唯一的操作系统

如果您打算完全删除该 Linux 系统，在备份好所有需要的数据后，使用任何分区工具（或任何系统的安装介质）将该系统所在的分区删除。至于 MBR 的引导信息，会在下次安装系统时被覆盖（也可使用分区工具立即清除）。

### 2. Red Hat 企业版 Linux 与其他系统共存形成双（多）引导

如果您要完全删除所有系统，参照上一步执行即可。如果要删除 RHEL 的系统，而保留其他系统，删除 RHEL 的相应分区，然后还要恢复其他系统的引导。

另一个系统是 Windows，恢复 Windows 系统引导的方法为：利用 Windows 的安装盘进入 Windows 修复控制台，在提示符后输入命令 `fixmbr`。

另一个系统是 Linux，因为大部分 Linux 系统采用 GRUB 进行引导，使用 Linux 安装程序救援模式修复 GRUB MBR 信息即可（后续详细学习 GRUB 引导过程）。

### 3. 虚拟机中的 Linux

删除 Linux 系统的虚拟磁盘映象文件和虚拟机配置文件即可。根据您所使用的虚拟化平台方式有所不同。



## 本章总结

安装系统是部署一个业务应用的起点，也是应用架构的基础。所以提前合理的规划、对于安装细节的理解会使用整个工作得心应手，减少以后系统使用中不必要的麻烦。而能应对不同的机房环境、服务器设备、安装介质、安装要求，并能解决安装过程中出现的问题，则需要在以后的学习中，对 Linux 系统体系结构更加深入了解后，通过反复实践和经验的积累进行加强。





随着学习的深入和工作实践，以后在部署 Linux 系统时，要尽量考虑到安全性、稳定性、可扩展性，最大限度地发挥系统软硬件性能，以及最优的性价比等多方面的因素。

### 章节重点：

掌握 RHEL 系统的安装方法。



## 基础技能演练

安装 RHEL 6.x，要求：

- (1) 在虚拟机中安装 RHEL 6.x 32 位（或 64 位）系统。
- (2) 使用 CD-ROM 方式 ISO 介质安装。
- (3) 选择中文安装语言。
- (4) 自定义分区。
- (5) 安装桌面环境。
- (6) 其他安装选项选择默认或适当的值。



## 提高技能实践

1. 安装 RHEL 6.x，要求：

- (1) 在虚拟机中安装 RHEL 6.x 32 位（或 64 位）系统。
- (2) 使用 CD-ROM 方式 ISO 介质安装。
- (3) 选择英文安装语言。
- (4) 选择使用所有空间并默认分区的方式，勾选“查看并修改分区布局”复选框。
- (5) 安装桌面环境并选择“现在自定义”单选按钮，添加远程桌面客户端相关的软件包。
- (6) 其他安装选项选择默认或适当的值。

2. 使用文本模式安装 RHEL 6.x 系统。

3. 有条件的尝试在物理真机上安装 RHEL 6.x 系统。

4. 安装其他流行的 Linux 发行版，如 CentOS、Ubuntu、Fedora 等。



## 第3章

# GNOME图形桌面使用入门

GNOME1 是 Red Hat Enterprise Linux 的默认图形桌面环境。它在图形框架(由 X Window System 提供)之上为用户提供了颇具吸引力的集成桌面和统一开发平台。GNOME 桌面环境包含集成应用程序(如文件管理器、文本编辑器等),以及系统图形管理工具。

要使用 GNOME 桌面环境,必须在系统安装时勾选相应的软件组(或安装完系统后额外安装)。Linux 系统管理以其功能强大且灵活的命令行界面为特色,对于多数服务器应用场景,图形界面是不需要的,且管理员通常选择不安装图形界面以节省系统资源。但我们仍有必要了解 Linux 图形环境,原因是有时候使用图形工具能更轻松、更标准地完成某些操作,而且系统管理员也可能要为图形用户提供支持。



### NOTE

除了 GNOME, Linux 系统采用的桌面环境还有 KDE、XFCE、LXDE、Unity 等。

## 3.1 登录到RHEL的GNOME桌面

当启动 RHEL 6 系统完成后,系统停留在等待用户登录的界面,如图 3-1 所示。

该界面相应的组件功能指引用户的下一步操作。

### 1. 登录窗口

选择需要登录的用户 Icon,单击后弹出要求输入密码的输入框,输入正确的密码后按 Enter 键即可登录系统。登录窗口的“其他”选项提供用户手动输入不在用户列表中显示的用户名的功能(root 用户默认不在列表中显示)。

1 本文关于GNOME的术语描述大多来自于GNOME的帮助文档,您可以通过访问“系统”→“帮助”来获得GNOME帮助文档的详细内容。



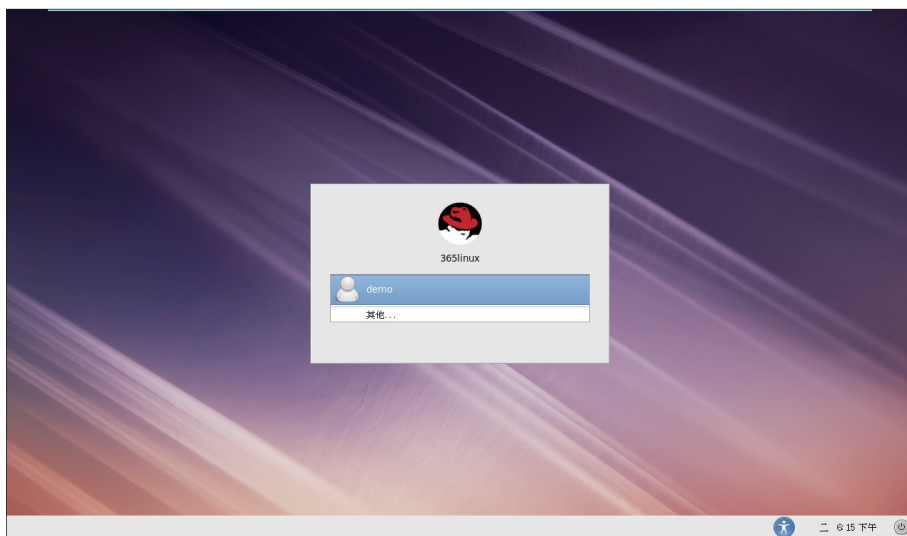


图3-1 用户登录界面

## 2. 底部面板

底部面板分别是“环境语言选择（选择用户后出现）”、“通用访问首选项”、“当前日期”、“系统关机或重启按钮”。

用户登录成功后，则进入到如图 3-2 所示的 GNOME 桌面环境。

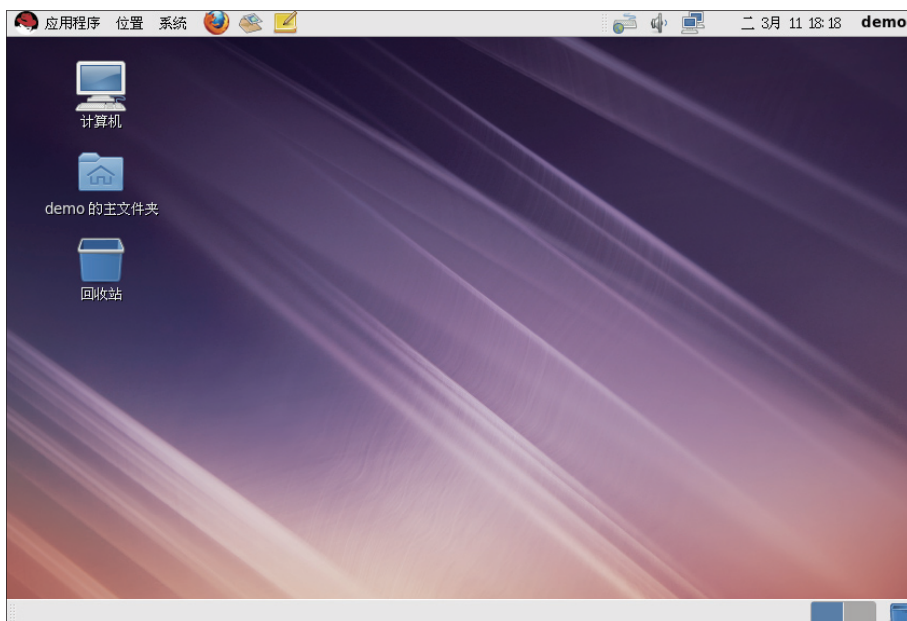


图3-2 GNOME桌面



## 3.2 配置RHEL 6的显示分辨率和多显示器

### 1. 配置 RHEL 6 的显示分辨率

首先配置显示分辨率，以达到最优的屏幕显示效果。通常 RHEL 6 系统安装好后会自动识别显卡和显示器，并设置最佳的分辨率，但仍然允许用户自己调整合适的屏幕分辨率。单击“系统”→“首选项”→“显示”，打开“显示首选项”对话框，如图 3-3 所示，可以在该对话框中调整分辨率。

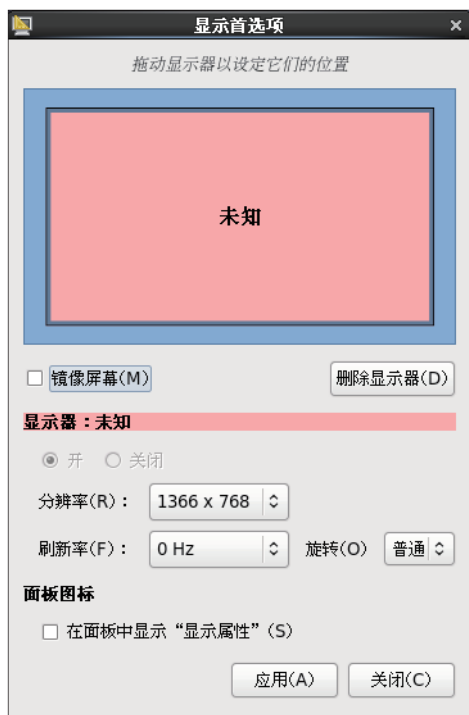


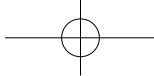
图3-3 显示首选项

### 2. 改变显示器的方向

在图 3-3 所示的“显示首选项”对话框中提供了旋转显示器的设置，可以通过旋转来设置屏幕显示的方向。这对运行一些倒置放置的显示器有用。

### 3. 配置多个显示器

RHEL 6 允许将屏幕显示投影（扩展）到多个显示器上。当连接到多个显示器时，系统会自动显示多个显示器，拖动显示器的标识来设定它们的前后位置。如果要使两个显示器的内容完全一样，需要在图 3-3 中勾选“镜像屏幕”复选框。



## 3.3 GNOME桌面概述

当您第一次进入桌面会话时，您将看到默认的启动屏幕，以及面板、窗口和各种图标。GNOME 桌面主要的组成如下。

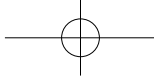
- 桌面。桌面在所有其他组件之下。您可以在桌面放置启动器对象，以便快速访问您的文件和文件夹，或者打开您经常使用的应用程序。
- 面板。面板是两个长条，分别在屏幕的上边和下边。默认情况下，上面板中显示 GNOME 的主菜单栏、日期和时间，以及 GNOME 帮助系统启动器；而底部面板显示打开窗口列表和工作区切换按钮。面板可以通过定制来包含不同的工具，像一些其他菜单和启动器，小工具应用程序，称作面板小程序。例如，您可以设置您的面板来显示当地的天气情况。
- 窗口。大多数应用程序运行在一个或几个窗口中。您可以在桌面上同时显示多个窗口。窗口可以根据您的使用情况来改变大小、移动位置。每个窗口的顶部都有一个标题栏，上边有最小化、最大化、关闭按钮。
- 工作区。您可以在工作区里再划分桌面，每个工作区可以包含许多窗口，允许您归类相关的任务。
- 文件管理器。通过 Nautilus 文件管理器，您可以访问您的文件、文件夹和应用程序。在文件管理器中，您可以管理文件夹的内容，打开文件。
- 控制中心。您可以在控制中心里定制您的计算机，它可以在面板顶部的系统菜单里找到。控制中心里的各个首选项工具允许您更改系统的某一部分。

无论普通用户还是系统管理员，都可以根据自己的需要改变设置，因而每个用户的桌面呈现的样式可能不一样。

### 1. 桌面

在屏幕里，桌面在所有组件的最下面。当没有窗口显示时，桌面是上下两个面板条之间的部分。您可以在桌面上放置文件夹和文件，以方便快速访问它们。桌面上有几个特殊对象。

- 计算机图标：可以让您访问光驱、软盘类的可移动介质，以及整个文件系统（也就是根文件系统）。
- 主文件夹：“用户”的主文件夹，这里存放您的个人文件。您可以从“位置”菜单打开自己的主文件夹。
- 回收站：是一个特殊文件夹，里面存放您不再需要的文件。
- 当您插入一个 CD 光盘、一个闪存，或者其他可移动设备，或者音乐播放器、数码相机时，桌面上会显示相应的设备图标。



当您使用计算机的时候,桌面可能会被各个窗口挡住。要快速显示桌面,最小化所有窗口,您可以使用下面的操作:

- 单击底部面板最左边的显示桌面按钮。
- 按组合键 **Ctrl+Alt+D**。

**NOTE**

您放在桌面上的文件或文件夹,将被放在您的主文件夹里的一个特殊的文件夹中,称作“桌面”。像其他文件夹一样,您可以往里面存放文件(或文件夹),唯一不同是这些文件或文件夹将显示在桌面上。

在桌面的任何空白区域单击鼠标右键,弹出右键菜单,如图 3-4 所示。

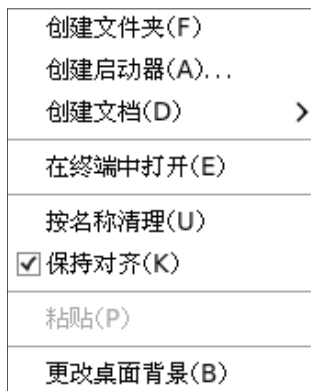


图3-4 桌面右键菜单

- 创建文件夹:在桌面新建一个文件夹。
- 创建启动器:在桌面上放置应用程序的启动图标,双击该图标即可启动相应的程序。  
在桌面上放置程序启动图标的另一个方法是,在“应用程序”菜单中找到相应的程序启动菜单,单击鼠标右键,在弹出的快捷菜单中选择“将此启动器添加到桌面”命令。
- 创建文档:在桌面上创建一个新的空文档。
- 在终端中打开:启动一个 **Shell** 命令行终端窗口。该终端 **Shell** 的当前工作目录将为用户的主文件夹的桌面子目录。
- 按名称清理:按字母顺序排序和组织排列桌面图标。
- 保持对齐:是否允许桌面图标随意摆放或必须对齐。
- 更改桌面背景:用户修改桌面背景图片。

在桌面图标上单击鼠标右键会弹出对具体对象的操作菜单,如图 3-5 所示。

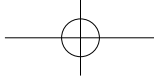


图3-5 文件右键菜单

## 2. 面板

在桌面的顶部和底部的两个长条称为面板，面板上的内容和内容的位置可以改变，顶部面板显示如图 3-6 所示。



图3-6 顶部面板

默认情况下，顶部面板上包含以下部分。

- 菜单栏：面板的菜单栏包括应用程序、位置及系统菜单。
- 一组程序快速启动器图标：图标的多少依赖于您的 GNOME 发行版，但通常您至少会看到一个 Web 浏览器启动器、一个 E-mail 客户端和一个记录便签。单击任何一个都可以启动相应的程序。
- 通知区域：显示系统的通知提醒，如软件更新、root 授权等。
- 系统设置图标：如输入法、音量控制、网络连接图标等。
- 时钟：显示当前时间，单击时间可以打开一个小日历。
- 用户账号信息：显示当前桌面会话登录的用户。

在默认情况下，底部面板包含 3 个对象，如图 3-7 所示。



图3-7 底部面板



- 窗口列表：为每个打开的窗口显示一个按钮。窗口列表小程序允许您最小化和恢复窗口。
- 工作区切换：允许您在工作区之间切换。
- 垃圾桶：已删除文件的回收站。

### 3. 快捷键

在 RHEL 系统中提供了大量桌面环境的快捷键，可以通过键盘的组合操作来快捷地完成一些对窗口或工作区的操作。可以通过单击“系统”→“首选项”→“键盘快捷键”，来调出系统默认快捷键的设置。

在弹出的如图 3-8 所示的对话框中，除了可以查看快捷键设置，还可以修改默认的键值、禁用某些对象的快捷键，或者添加自定义的快捷键。

常用的快捷键如 Ctrl+Alt+L、Alt+F2 等。

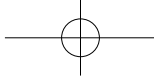


图3-8 键盘快捷键设置

## 3.4 结束当前会话

当您使用完自己的计算机，可以选择下面的动作之一：

- 注销以允许其他用户使用。要从 GNOME 注销，单击“系统”→“注销用户名”。



- 关闭计算机并切断电源。要关闭计算机，单击“系统”→“关机”，然后在弹出的对话框中单击“关机”按钮。

## 3.5 文件管理操作

Nautilus 是 GNOME 桌面附带的文件管理器。通过 Nautilus，您可以了解文件系统、创建文件和文件夹、查看文件属性，以及处理文件和文件夹（复制、删除、移动、剪切、粘贴等）。

### 1. 改变 Nautilus 的浏览模式

默认情况下，Nautilus 是在单窗口模式下进行操作，每次打开一个文件夹时，都会打开一个新窗口，如图 3-9 所示。您也可以将其更改为浏览器模式，每次更改新文件夹时，当前文件夹会立刻更改为打开的文件夹的内容，而不会打开新窗口，如图 3-10 所示。要对当前用户使用浏览器模式，依次单击“系统”→“首选项”→“文件管理”。在“文件管理首选项”对话框的“行为”选项卡，勾选“总在浏览器窗口中打开”复选框（或者在打开的文件夹窗口菜单上单击“编辑”→“首选项”也可打开“文件管理首选项”对话框），如图 3-11 所示。



图3-9 文件管理器窗口

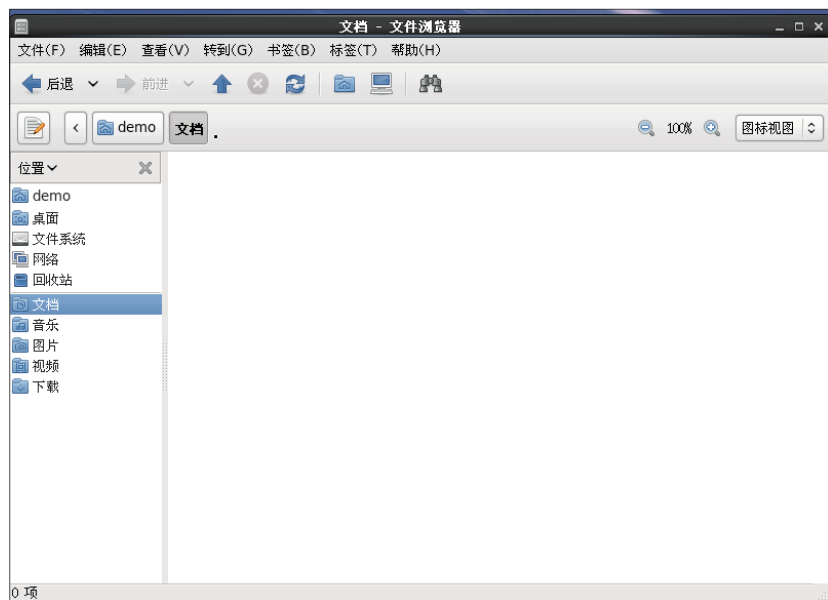


图3-10 文件浏览器

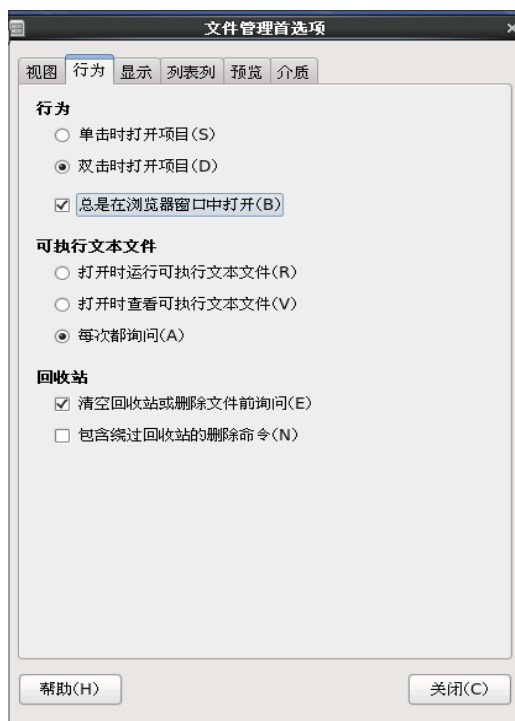
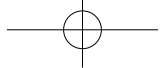


图3-11 “文件管理首选项”对话框

## 2. 使用 Nautilus 管理本地文件夹

在默认情况下，文件和文件夹由图标与文件名表示（图标视图）。单击“查看”下拉菜





单可将显示方式更改为“列表视图”或“紧凑视图”。

文件名以句点（.）开头的文件是指隐藏文件。选中“查看”→“显示隐藏文件”选项，来显示出当前文件夹中的隐藏文件或文件夹。

Linux 的文件系统层次结构遵从 FHS 标准，有着统一的目录组织结构，但实际上各个发行版会略有差异。

文件系统层次结构标准（Filesystem Hierarchy Standard, FHS）定义了 Linux 操作系统中的主要目录及目录内容。FHS 由 Linux 基金会维护，这是一个由主要软件或硬件供应商组成的非营利组织，例如 HP、Red Hat、IBM 和 Dell。多数 Linux 发行版遵从 FHS 标准，并且声明其自身政策以维护 FHS 的要求。

在 FHS 中，所有的文件和目录都出现在根目录“/”下，即使它们存储在不同的物理设备中，如图 3-12 所示。但是请注意，这些目录中的一些可能或可能不会在 UNIX 系统上出现，这取决于系统是否含有某些子系统，例如 X Window 系统的安装与否。

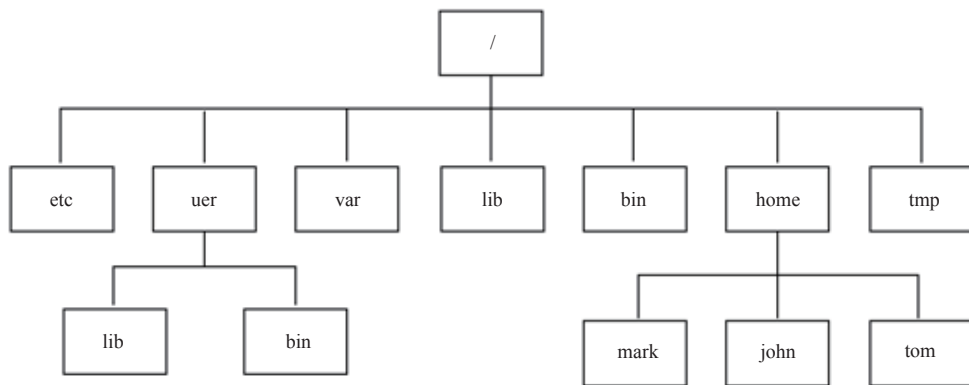


图3-12 Linux文件系统结构

RHEL 6 中根目录下的第一层目录解析如表 3-1 所示。

表3-1 根目录解析

目 录	说 明
/bin	用于存放系统命令的目录之一。其特点是/bin 存放的是在单用户模式下使用的命令，面向所有用户
/boot	存放引导程序文件，例如 kernel、initrd。通常需要单独的分区
/dev	设备文件目录。在 Linux 系统上，任何设备都以文件的形态呈现
/etc	系统配置文件目录
/home	普通用户的家目录。一般为单独的分区
/lib	/bin 和 /sbin 中二进制文件必要的库文件。还包含系统启动时的库文件
/lost+found	这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件
/media	自动加载的可移动设备挂载目录
/misc	AutoFS 服务挂载目录
/mnt	临时挂载目录。一般在手动挂载时使用



续表

目 录	说 明
/net	AutoFS 服务挂载目录（用于使用 IP 访问共享主机所有的共享目录）
/opt	可选的第三方软件安装存放目录
/proc	虚拟文件系统，将内核与进程状态归档为文本文件存放
/root	超级用户的家目录
/sbin	必要的系统命令存放目录。设计只有 root 用户才能运行系统命令
/selinux	增强安全的 SELinux 的单独目录
/srv	某些网络服务取用数据目录
/sys	虚拟文件系统，主要记录与内核相关的信息，比如设备信息和内核模块信息
/tmp	临时目录。设计为系统重启后目录中的文件会被删除
/usr	UNIX Software Resource 的缩写，存放系统组件和应用程序的主要目录
/var	主要存放经常变动的文件，比如日志、缓存，以及服务的应用数据等

**NOTE**

作为前瞻，再通过教师的解释或查阅资料了解挂载点（挂载目录）的概念。

### 3. 在 Nautilus 中访问远程文件系统

如果有网络连接，Nautilus 可以显示可远程访问的 FTP、SSH、Windows 共享及其他类型服务器中的文件和文件夹。单击菜单“位置”→“连接到服务器”，在弹出的对话框中选择服务器的类型，并填写相应的验证信息（或匿名）登录。

以 SSH 方式远程访问另一台服务器的文件夹为例。

**Step 01** 在“连接到服务器”对话框的“服务类型”下拉列表框中选择“SSH”，其中“服务器”文本框中填写您要访问的远程服务器的 IP 地址，如图 3-13 所示的“192.168.148.132”；“用户名”文本框中填写您被远程服务器授权访问的账号信息。单击“连接”按钮。

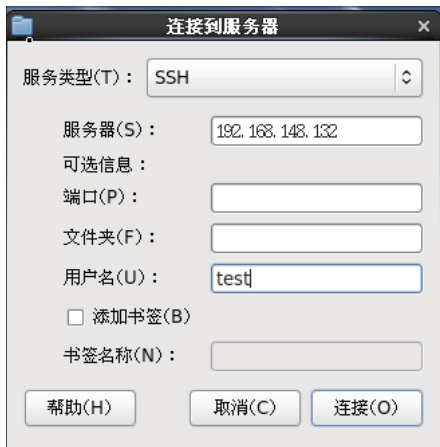


图3-13 连接到服务器



**Step 02** 在弹出的对话框中输入 test 用户的密码，如图 3-14 所示，单击“连接”按钮继续。



图3-14 输入密码

连接成功后，在“文件浏览器”窗口中会显示在远程服务器“192.168.148.132”上的文件，效果如图 3-15 所示。

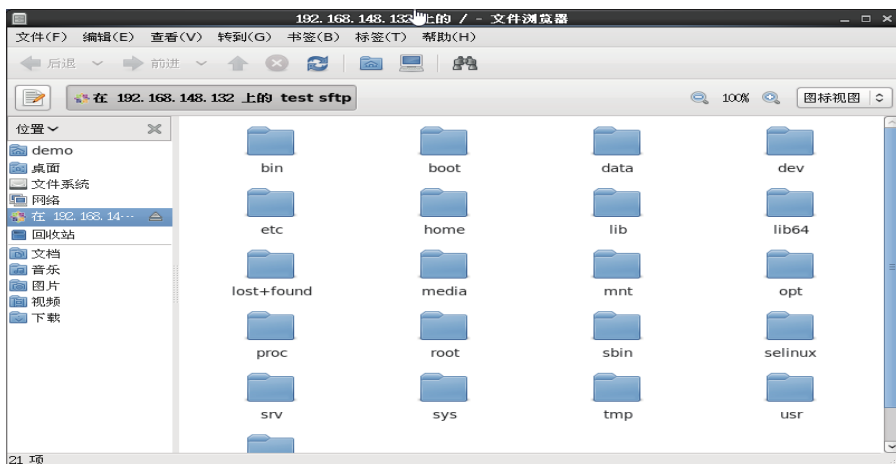


图3-15 访问远程文件系统

## 3.6 通过gedit编辑文件

Linux 的设计原则之一是配置文件基于文本。使用简单的文本编辑器就可以对大多数程序进行设置、修复等工作，而不需要特殊的配置工具。因此了解如何在 Linux 中编辑文本文件非常重要。

gedit 文本编辑器是用于编辑文本的图形工具。通过单击“应用程序”→“附件”→“gedit 文本编辑器”可以启动 gedit 编辑器窗口，如图 3-16 所示。



图3-16 gedit工作窗口

## 3.7 使用配置工具查看和更改系统时间

任何在运行中的系统都要保证系统时间的准确。RHEL 可以通过手动调整时间或通过 Network Time Protocol (NTP) 与时间服务器同步时间，同时需要设置正确的系统时区。设置系统时间可以通过图形界面“日期和时间属性”工具设置，也可以通过命令行进行设置，这里我们先介绍图形界面下的操作。

**Step 01** 单击“系统”→“管理”→“日期和时间”打开系统时钟配置工具。因为普通用户没有对系统修改的权限，所以，除非您已经进行过认证授权，否则运行时需要输入管理员的密码进行授权，如图 3-17 所示。

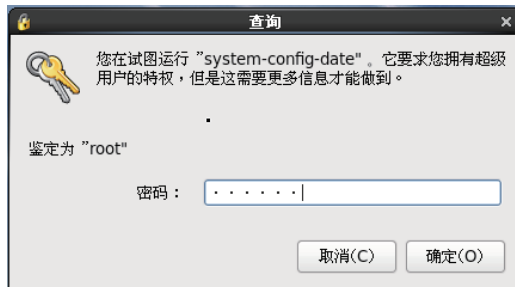


图3-17 需要root授权

**Step 02** 在图 3-18 所示的“日期 / 时间属性”对话框中手动调整日期和时间。

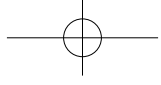


图3-18 设置日期和时间

如果勾选图 3-19 中的“在网络上同步日期和时间”复选框，则表示使用 NTP 与时间服务器进行时间同步。RHEL 提供了一组官方的时间服务器列表。与官方网络时间服务器同步需要连接外部网络的支持。可以单击“添加”按钮，添加自定义的时间服务器（比如内网的时间服务器）。



图3-19 在网络上同步时间



**Step 03** 单击图 3-20 所示的“时区”选项卡，配置正确的时区。如果您的系统时钟使用 UTC 时间，则勾选底部相应的复选框。UTC 是指协调世界时，又称世界标准时间或世界协调时间，是最主要的世界时间标准，其以原子时秒长为基础，在时刻上尽量接近于格林尼治标准时间 (Greenwich Mean Time, GMT)。本地时间则由 UTC 时间加上或减去时区时差得到，比如中国大陆就称 UTC+8。



图3-20 设置时区

**NOTE**

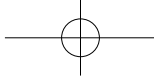
使用命令行对时间的设置将在后面的章节涉及。

**REAL-WORK TIP**

理解实例中的 root 管理权限在应用中的授权动作。

**本章总结**

本章节主要带领大家了解 Linux 桌面环境 GNOME 的工作方式。在 GNOME 的桌面环境中还有大量的实用工具和应用程序，能帮助我们更轻松地使用和管理系统。虽然 Linux 有强大的命令行功能，且对于熟练的系统管理员来说，使用命令行来控制系统更为直接，但在必



要时，使用图形界面配合更加有效。如果您刚开始使用 Linux，使用图形界面对了解 Linux 系统的某些概念很有帮助。

#### 章节重点：

- 认识 RHEL 系统的 GNOME 桌面环境。
- 能使用 GNOME 图形工具配置系统。
- 理解对于系统关键属性更改时需要管理员权限。



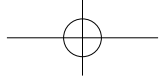
## 基础技能演练

1. 进入 GNOME 桌面。
  - (1) 使用普通用户和其密码登录。
  - (2) 锁定屏幕的操作。
  - (3) 注销当前用户。
  - (4) 重新进入桌面。
  - (5) 关闭系统。
2. 改变桌面背景。
3. 使用 gedit 编辑器在用户主文件夹中创建一个文本文件（如 student.txt），并编辑该文件的内容后保存。
4. 使用 Nautilus 管理本地文件。
  - (1) 打开用户的主文件夹，在该目录下创建一个名为 targetdir 的文件夹。
  - (2) 将文件系统下，boot 目录 /grub 目录下的 device.map 复制到刚刚创建的 targetdir 目录中。
  - (3) 在用户的主文件夹中创建指向 targetdir 目录下的 device.map 的链接文件，打开链接文件查看文件内容。
  - (4) 删除实验新建（复制）的文件和文件夹，并从回收站彻底删除。
5. 设置系统的时间和时区。



## 提高技能实践

1. 修改当前用户的密码。
2. 创建一个新的用户，并在两个用户间进行切换登录。
3. 配置系统的网络连接，能够访问同一网段内的其他机器（或能访问外网）。
4. 使用 Nautilus 访问远程共享文件夹（如网络内的 FTP 服务器，或 SSH 其他 Linux 服务器）。
5. 阅读系统图形界面下的帮助文档或访问 Red Hat 在线文档获得帮助解决上述问题。



## 第4章

# Bash入门

在学习和使用 Linux 系统的过程中，Shell 命令行是非常重要的组成部分。虽然图形桌面环境如 GNOME 提供了一个友好的操作界面，但在实践中，Shell 命令行提供了更多的功能、更好的灵活性，以及自动化和批量处理的能力，可以简化或实现那些使用图形工具难以有效完成的操作。Shell 环境还提供了在 Linux 服务器中无法使用图形界面交互的可能，例如在系统非正常启动的情况下，或基于字符界面的远程连接管理。

Red Hat Enterprise Linux 中的默认 Shell 是 Bash。

## 4.1 什么是Shell

Linux (or UNIX) Shell 也叫作命令行界面，它是 Linux/UNIX 操作系统下传统的用户和计算机的交互界面，用户可直接输入命令来执行各种各样的任务。普通意义上的 Shell 就是可以接受用户输入命令的程序。之所以被称作 Shell，是因为它隐藏了操作系统底层的细节。Linux 操作系统下的 Shell 既是用户交互的界面，也是控制系统的脚本语言。

在 Linux 系统中被广泛使用的 Shell 是 Bash，在 1987 年由布莱恩·福克斯 (Brian J. Fox) 为了 GNU 计划而编写。1989 年发布第一个正式版本，原先是计划用在 GNU 操作系统上，但能运行于大多数类 UNIX 系统的操作系统之上，包括 Linux 与 Mac OS X v10.4 都将它作为默认 Shell。在 Novell NetWare 与 Android 上也有移植。

除此之外，Shell 还有 ash、dash、ksh、zsh、csh、tcsh 等。

## 4.2 使用Shell命令行

默认 RHEL 6 提供了 7 个虚拟控制台，可以使用 Ctrl+Alt+F1~F7 组合键来进行切换，其





中控制台 1（或 7）是图形界面，其余为命令行控制台，如图 4-1 所示。

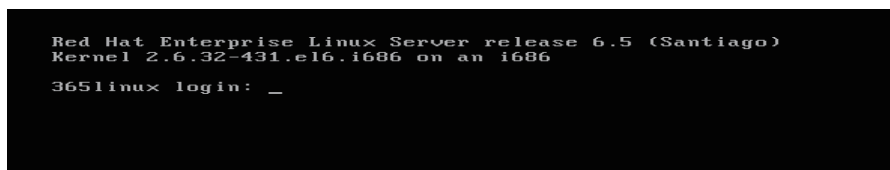


图4-1 虚拟控制台

在桌面环境下，可以使用虚拟终端来运行命令。GNOME 中的默认终端模拟器是 `gnome-terminal`。可以通过如下方式启动：

- 从顶部面板依次单击“应用程序”→“系统工具”→“终端”打开。
- 在桌面空白区域或 Nautilus 窗口空白区域单击鼠标右键，在弹出的快捷菜单中选择“在终端中打开”命令。
- 按快捷键 `Alt+F2`，在“运行应用程序”对话框中输入“`gnome-terminal`”，回车运行，如图 4-2 所示。



图4-2 运行应用程序

通过以上任一种方式打开虚拟终端窗口，如图 4-3 所示。

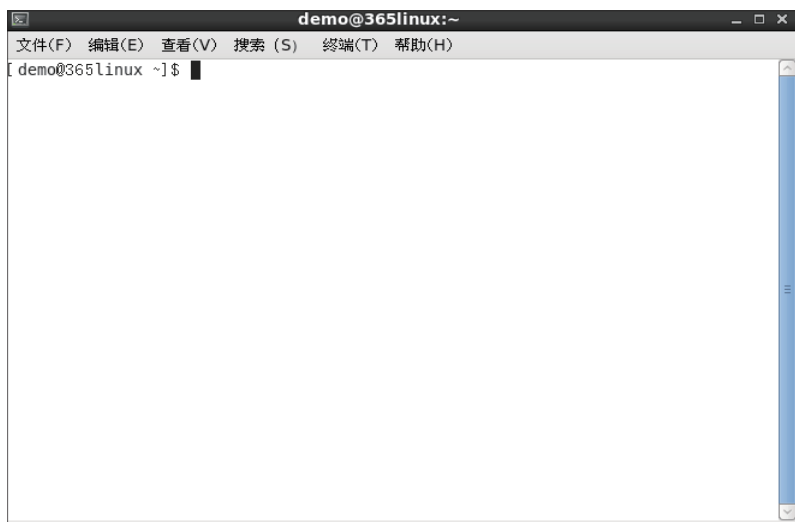


图4-3 虚拟终端窗口



在 Shell 提示符终端输入命令。如图 4-3 所示，标准提示符列出了当前用户的登录名称、计算机的较短主机名、当前目录的名称（用方括号括起来）、后跟 \$ 提示符。如果以超级用户（root）身份运行 Shell（参见下文示例），\$ 将替换为 #，以便更明显地说明这是超级用户在工作。

用户可以从命令控制台、虚拟终端、远程客户端登录系统并执行命令。

## 4.3 在命令行中启动图形工具

就像任何其他程序一样，您可以从命令行启动包含图形界面的程序。例如，您可以在一个图形虚拟终端窗口中的 Shell 提示符下输入 `firefox` 命令启动图形的 Web 浏览器。

**示例：**在命令行中启动 Firefox 浏览器。

```
[demo@365linux ~]$ firefox
```

但是，这种做法的缺点是只要图形程序仍然在运行，用于启动的 Shell 提示符就会被占用而一直不可用，这种情况称为程序在前台运行。为了避免这种不便，可以在提示符下的命令行末尾处添加一个 `&`，以在后台启动程序。

**示例：**使用后台运行的方式在命令行中启动 Firefox 浏览器。

```
[demo@365linux ~]$ firefox &
```

Bash 还提供了通过 Shell 提示符更改进程的运行方式（前台、后台）。

- 在 Bash 中可以使用 `Ctrl+C` 组合键终止前台进程。
- 使用 `Ctrl+Z` 组合键暂停前台进程并返回 Shell 提示符。
- 在终端中执行 `jobs` 命令列出该与该 Shell 相关联的在后台运行或已停止的进程。
- 在终端中使用 `fg` 命令可以向前台发送作业。
- 在终端中使用 `bg` 命令可以运行后台暂停的进程。

**示例：**进程的前后台切换。

**Step 01** 在前台打开 `gedit` 文件编辑器。

```
[demo@365linux ~]$ gedit
```

**Step 02** 按 `Ctrl+Z` 组合键将进程暂停。

```
^Z
```

```
[1]+  Stopped                  gedit
```

**Step 03** 查看该 Shell 终端的后台或已暂停的进程。

```
[demo@365linux ~]$ jobs
```

```
[1]+  Stopped                  gedit
```

**Step 04** 使用 `bg` 命令将进程在后台恢复运行。

```
[demo@365linux ~]$ bg 1
```



```
[1]+  gedit &
```

**Step 05** 使用 `fg` 命令将进程发送至前台，并使用 `Ctrl+C` 组合键结束进程。

```
[demo@365linux ~]$ fg 1
gedit
^C
```



#### REAL-WORK TIP

通常在 Shell 终端提示符中运行的作业（非服务进程）是和该 Shell 相关联的，当该 Shell 终端被关闭时，运行的作业也会停止，这时可以使用 `nohup` 命令来脱离这种关联性。

如果出于某些原因您需要以 `root` 用户身份运行图形程序，而 `PolicyKit`（普通用户特权获取机制管理）不支持以普通用户身份运行该图形程序时，在必要时进行 `root` 用户的授权（比如 `Nautilus` 文件管理器），那么则需要在命令行下切换到 `root` 用户，然后在命令行中以 `root` 的身份打开该图形程序。

**示例：**以 `root` 用户身份打开 `Nautilus` 文件管理器。

```
[demo@365linux ~]$ su -
密码：
[root@365linux ~]# nautilus &
[root@365linux ~]# exit
```



#### REAL-WORK TIP

在命令后中使用 `su -` 命令切换用户后，想要回到之前的登录用户，要使用 `exit` 命令退出当前用户，则退回到之前的用户，而不能使用 `su -` 来回反复切换。

## 4.4 命令的格式

在 Shell 提示符下面输入的命令由三部分组成：

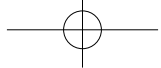
命令	选项	参数
command	[-options]	[parameter1] [parameter2] ...

**示例：**执行 `ls` 命令，其中，`-l`、`-r` 是短选项，`--size` 是长选项，`/boot` 是命令执行的参数。

```
[demo@365linux ~]$ ls -l --size -r /boot
```

Bash 命令至少有如下特点：

- 一条命令必须是以可执行的命令开头，以空格隔开；选项是可选的，但在大多数情况下被使用以满足用户的功能定制要求；参数一般是命令要操作的对象，一条命令的参数可以是一个或多个，没有参数则取命令参数默认值。输入完命令后按 `Enter` 键执行。



- “-”后面接简写的选项（字母），可以把多个简写的选项串在一起，不过有时要注意顺序；长选项（由单词或单词缩写组成）用“--”分隔。选项也可以有自己的参数，如“--width=40”。
- 命令中间的空格不论几格，Bash 都视为一格，命令太长时用“\”转义“Enter”，使用命令可以换行继续输入。
- Linux 命令严格区分大小写。
- 在大部分情况下，对于命令执行的结果，没有输出意味着命令执行成功。
- Tab 键补全命令和路径。
- Bash 具有历史命令功能（history）。
- 命令可以使用别名的形式（alias）。
- 有大量的快捷键可以使用。
- 有完善的帮助文档。

以下对 Bash 的部分重要功能做详细的解释。

## 1. 命令行 Tab 键补全功能

Tab 键补全允许您在提示符下输入足够的内容以使其唯一后快速补全命令和文件名。如果输入的字符匹配到的命令或文件名不唯一，则按 Tab 键两次，以显示所有以输入的字符开头的命令或文件名的情况。Tab 补全的功能可以提高命令的输入速度，且可以判断输入的命令或文件名的正确性。

**示例：**使用两次 Tab 键显示出以 pas 开头的命令。

```
[demo@365linux ~]$ pas<Tab><Tab>
passwd      paste      pasuspender
```

**示例：**使用 Tab 键自动补全命令 passwd。

```
[demo@365linux ~]$ pass<Tab>
[demo@365linux ~]$ passwd
```

**示例：**使用 Tab 键自动补全路径。

```
[demo@365linux ~]$ ls /etc/sysco<Tab>/ipt<Tab>
[demo@365linux ~]$ ls /etc/sysconfig/iptables
```

## 2. Bash 历史记录功能

Shell 历史记录允许您查看之前运行过的命令并对其进行编辑或再次执行。使用 history 命令查看所有之前运行过的命令，或者使用上箭头和下箭头一次滚动浏览一个历史记录命令。历史记录命令的输出包含数字值。在感叹号（!）后面使用该数字可以再次运行该命令。在感叹号后面使用非数字值则运行最后一个以这些字符开头的命令。

**示例：**使用 history 命令显示当前会话所有的历史命令。

```
[demo@365linux ~]$ history
1  su -
```



```
2 cd
3 firefox &
4 su -
5 history
```

示例：执行编号为 3 的历史命令 `firefox &`。

```
[demo@365linux ~]$ !3
```

### 3. 使用命令别名

对于一些较长的命令执行格式或者命令组合，而又经常使用的，可以使用别名的方式进行定义，以减少反复较长的输入。使用 `alias` 命令可以显示和定义别名，使用 `unalias` 取消命令别名。除非将别名的定义写到用户的配置文件，否则别名只在当前会话中有效。

示例：显示当前会话所有别名。

```
[demo@365linux ~]$ alias
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot
--show-tilde'
```

示例：自定义别名 `p2`。

```
[demo@365linux ~]$ alias p2='ping -c 3 192.168.148.2'
[demo@365linux ~]$ p2
PING 192.168.148.2 (192.168.148.2) 56(84) bytes of data:
64 bytes from 192.168.148.2: icmp_seq=1 ttl=128 time=0.112 ms
64 bytes from 192.168.148.2: icmp_seq=2 ttl=128 time=0.386 ms
64 bytes from 192.168.148.2: icmp_seq=3 ttl=128 time=0.141 ms

--- 192.168.148.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.112/0.213/0.386/0.122 ms
```

示例：取消自定义的别名 `p2`。

```
[demo@365linux ~]$ unalias p2
[demo@365linux ~]$ p2
-bash: p2: command not found
```

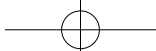


#### REAL-WORK TIP

用户常用的命令别名定义命令应该写到用户家目录下的 `.bashrc` 文件中，以保证每次用户登录都能使用该别名。

### 4. 命令的物理位置

Linux 命令分为两种类型。



- 内部命令：Bash 在本身内建的命令，这些命令在 Shell 启动时被加载到内存。
  - 外部命令：内建命令之外的可执行程序，通常是由系统的组件或者应用程序安装提供。
- 通过 `which` 命令可以定位命令在系统中的真实路径。

示例：使用 `which` 命令定位 `ls` 命令的路径。

```
[demo@365linux ~]$ which ls
alias ls='ls --color=auto'
/bin/ls
```

示例：使用 `which` 命令定位 `history` 命令，结果显示 `history` 并不存在于系统命令路径中，可见 `history` 命令是 Bash 内建的命令。

```
[demo@365linux ~]$ which history
/usr/bin/which: no history in (/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin: \
/usr/local/sbin:/usr/sbin:/sbin:/home/demo/bin)
```

示例：显示 Bash 在系统中查找命令路径的环境变量。

```
[demo@365linux ~]$ echo $PATH
/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/demo/bin
```

## 4.5 Bash命令行的快捷键

RHEL 系统 Bash 命令行环境有大量的快捷键可以使用，方便操作。这里仅列出常用的快捷键，如表 4-1 所示。

表4-1 Bash快捷键

快捷键	功 能
Ctrl + c	非常规中断，终止前台进程。例如：中断命令 <code>cat /dev/zero</code>
Ctrl + d	输入完成的正常信号。例如：命令 <code>wc</code> 和 <code>at</code> 的操作
Ctrl + z	挂起前台进程，用 <code>fg</code> 恢复，用 <code>bg</code> 恢复后台执行
Ctrl + l	清屏，等同于命令 <code>clear</code>
Ctrl + u	删除当前行
Ctrl + h	删除光标前一个字符
Ctrl + w	清除光标之前的字符串
Ctrl + k	清除光标之后的字符串
Ctrl + a	光标移动到命令行的行首
Ctrl + e	光标移动到命令行的行尾
Ctrl + r	从历史命令中找含有输入字符的命令
Alt + b	光标往前移动一个字符串
Alt + f	光标往后移动一个字符串

快捷键	功 能
Ctrl + y	恢复 Ctrl + w 或 Ctrl + k 清除的内容
Ctrl + B	光标往前移动一个字符
Ctrl + F	光标往后移动一个字符
Ctrl + X	在光标所在位置与行首切换
Alt + .	补全之前输入过的参数
Alt + U	换成大写
Ctrl + S	锁住终端输出
Ctrl + Q	解锁终端

**REAL-WORK TIP**

在某些情况下（比如使用 `cat` 命令查看了不能直接打开的文件），会导致终端显示乱码并无法正常输入，此时可盲打 `reset` 命令进行终端复原。

## 4.6 获得命令的帮助信息

只了解命令单一的作用是不够的。为了有效地使用命令，还需要了解每个命令可以接受哪些选项和参数，以及命令希望如何排列这些选项和参数（命令的语法）。一个完整的系统里包含了数千个命令，而每个命令都有自己众多不同的选项和参数，这样庞大繁多的命令用法极大地体现了 Linux 命令行的灵活性，同时也增加了使用者的学习和应用难度。

然而几乎所有的命令和配置文件的语法格式与用法都有帮助文档。我们可以通过以下几种方式获得帮助信息：

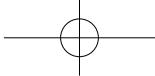
- 使用命令 `-h` (`--help`) 或者 `-?` 的选项来获得命令使用的规范和选项、参数的信息。
- 使用 `man` 来获得命令的使用手册。
- 使用 `pinfo` 读取文档。
- 查看 `/usr/share/doc` 中的文档。

### 1. `--help` 帮助输出

大多数命令都有 `-h`（有的命令 `-h` 有其他特定的功能，则只能使用 `--help`）的帮助选项，执行命令的该选项时会在终端输出简洁的帮助信息。

**示例：**使用 `--help` 选项获取 `ls` 命令的帮助信息。

```
[demo@365linux ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort.
```



```
Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
...
```

关于帮助输出的几个基本惯例：

- 方括号 ([ ]) 中的任何内容都为可选。
- 省略号 (...) 表示此字符串可以是任意长度。
- 以竖线 (|) 分隔的多个选项，表示可以选择其中任意一项。
- 尖括号 (< >) 中的文本表示必须出现变量数据，因此 <filename> 表示“在此插入要使用的文件名”。

## 2. 使用 man 读取帮助文档

Linux man 手册提供了比 help 输出更为详尽的帮助文档，类似于一本分成许多章节的大型书籍。用户可以通过在终端执行 man 命令来获取相关命令或文件的帮助文档信息。终端以每次一个屏幕的形式显示内容，并可通过键盘命令来控制导航 man 手册。man 导航按键如表 4-2 所示。

表4-2 导航man手册

按键 / 命令	结 果
空格键	向前滚动一个屏幕
方向键下	向前滚动一行
方向键上	向后滚动一行
/string	在 man 手册中向前搜索 string
n	在 man 手册中重复之前的向前搜索
N	在 man 手册中重复之前的向后搜索
q	退出 man 并返回到终端提示

示例：查看 ls 命令的 man 手册。

```
[demo@365linux ~]$ man ls
LS(1)                User Commands                LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
...
```

man 文件主要包括以下几个部分（各命令可能有区别）。

- NAME：程序名和简介。
- SYNOPSIS：命令的格式，显示所有的选项和参数。
- DESCRIPTION：命令功能的描述和选项的详解。
- OPTIONS：所有选项清单和描述。





- EXAMPLES : 用法举例。
- AUTHOR : 作者。
- REPORTING BUGS : 报告 bug。
- CCOPYRIGHT : 许可证。
- SEE ALSO : 相关内容。

man 手册分为多个章节, 所以可能会出现相同名称的多个帮助文档内容。为了区分不同的 man 手册, 在编写对 man 手册的引用时, 通常在 man 手册的名称后面添加手册的章节号并用括号括起来。比如, 用户命令 passwd 的帮助文档是 passwd(1), 存储本地用户信息的配置文件 /etc/passwd 的帮助文档是 passwd(5)。

关于 man 手册的章节, 可以通过 man man 来查看相关信息, 如表 4-3 所示。

表4-3 man手册的章节

章 节	man 手册类型
1	用户命令
2	内核系统调用 (从用户空间到内核的进入点)
3	库函数
4	特殊文件和设备
5	文件格式和规范
6	游戏
7	规范、标准和其他页面
8	系统管理命令
9	Linux 内核 API (内核调用)

man 命令会按特定顺序搜索手册中的各节, 并显示找的第一个匹配项。例如 man passwd 默认情况下将显示 passwd(1)。要找到特定节中的 man 手册, 则必须在命令行中以参数的形式指定节号, 例如 man 5 passwd 将显示 passwd(5)。

**示例:** 查看 /etc/passwd 配置文件的帮助文档。

```
[demo@365linux ~]$ man 5 passwd
```

可以使用 man -k keyword 对 man 手册执行关键字搜索, 这将产生一个相关 man 手册的列表, 包括对应的章节。

```
[demo@365linux ~]$ man -k passwd
chpasswd                (8)  - update passwords in batch mode
fgetpwent_r [getpwent_r] (3)  - get passwd file entry reentrantly
getpwent_r              (3)  - get passwd file entry reentrantly
gpasswd                 (1)  - administer /etc/group and /etc/gshadow
htpasswd                (1)  - Manage user files for basic authentication
kpasswd                 (1)  - change a user's Kerberos password
lpaswd                  (1)  - Change group or user password
lpasswd                 (1)  - add, change, or delete digest passwords
```



```
pam_localuser      (8) - require users to be listed in /etc/passwd
pam_passwdqc       (8) - Password quality-control PAM module
passwd             (1) - update user's authentication tokens
passwd2des [xcrypt] (3) - RFS password encryption
passwd             (5) - password file
...
```

**NOTE**

关键字搜索需要 `makewhatis` 更新数据库，通常系统每天会自动运行更新。

### 3. 使用 `pinfo` 读取文档

GNU Project 开发的软件使用 `info` 系统来提供其部分文档，`info` 文档通常以书籍的形式提供，其由包含超链接的 `info` 节点组成。此格式比 `man` 手册更加灵活，允许对复杂命令和概念进行更加彻底的说明。在某些情况下，某个命令同时存在相应的 `man` 手册和 `info` 文档；在大多数情况下，`info` 文档的信息更加详细。

示例：比较 `tar` 的 `man` 手册和 `info` 文档。

```
[demo@365linux ~]$ man tar
[demo@365linux ~]$ pinfo tar
```

### 4. `/usr/share/doc` 中的文档

如果在 `man` 手册、`info` 手册或者 GNOME 帮助实用程序都不能找到相关帮助文档，则可以在系统目录 `/usr/share/doc` 中查找。许多应用程序和系统命令的帮助文档位于该目录下以 RPM 软件包命令的子目录中。

示例：查看 `mdadm.conf` 配置文件的示例文件。

```
[demo@365linux ~]$ less /usr/share/doc/mdadm-3.2.6/mdadm.conf-example
```

## 4.7 命令使用示例

### 1. 从命令行管理文件

Linux 文件系统具有层次结构，其组织方式采用“倒树”模型。顶级目录称为根目录（/ 目录），是整个文件系统层次结构的起点，而根分区挂载到 / 目录。要在系统中指定文件的位置，可以指定该文件的绝对路径（从根目录到各级子目录到文件），或者使用相对路径（从当前工作目录到其下的各级子目录到文件）。

在命令行中文件的路径，如 `/usr/share/doc`，位于最前面的 `/` 表示根目录，即绝对路径的起点，之后的 `/` 则表示路径中目录的分隔符。

**NOTE**

在某些系统或说法中，经常将根目录 (/) 称作文件系统层次结构的 root (这里的 root 表示根的意思)，而系统中存在的 /root 目录是管理员用户 root 的家目录，容易造成混淆，一定要理解清楚。

### 1) 切换工作路径

**示例：**使用 `pwd` 命令查看当前的工作目录。

```
[demo@365linux ~]$ pwd
/home/demo
```

**示例：**使用 `cd` 命令切换工作目录。

**Step 01** 使用绝对路径方式进入 `doc` 目录。在命令行中，绝对路径作为参数一定是从根目录 (/) 开始，依次连接各级子目录。切换到目标目录后，终端提示符会改变成为当前目录的简写。

```
[demo@365linux ~]$ cd /usr/share/doc/
[demo@365linux doc]$
```

**Step 02** 使用相对路径方式进入当前 `doc` 目录下的 `zip-3.0` 目录。在命令行中，使用相对目录，即相对于当前的工作目录。使用相对目录时，要省略目录前的路径分隔符，否则会绝对路径产生混淆。示例中的 `zip-3.0` 等同于 `/usr/share/doc/zip-3.0`。

```
[demo@365linux doc]$ cd zip-3.0/
[demo@365linux zip-3.0]$
```

**Step 03** 返回上一级目录，参数 `..` 表示上一级目录；`.` 表示当前目录。

```
[demo@365linux zip-3.0]$ cd ..
[demo@365linux doc]$
```

**Step 04** 快速返回当前用户的家目录，参数 `~` 表示当前用户的家目录。“`~ zhangsan`”则表示用户 `zhangsan` 的家目录。

```
[demo@365linux doc]$ cd ~
[demo@365linux ~]$ pwd
/home/demo
```

**Step 05** 快速进入上一次工作目录。参数 `-` 表示切换到当前目录之前的目录。

```
[demo@365linux ~]$ cd -
/usr/share/doc
[demo@365linux doc]$
```

### 2) 查看目录文件列表

在 Linux 系统中，一个基本原则是“一切皆文件”，包括硬件设备。这样，通过简单的工具即可完成某些功能非常强大的操作。根据文件的特点，Linux 系统将文件分为 7 种类型。



- - : 一般文件。
- d : 目录。
- l : 链接文件。
- b : 块设备文件。
- c : 字符设备文件。
- s : 套接字文件。
- p : 管道文件。

符号或字母是在命令行中的标识符。

**示例：**用 `ls` 命令查看文件列表并显示文件属性（包括类型）。

```
[demo@365linux ~]$ ls -l /boot
总用量 26149
-rw-r--r--. 1 root root 109953 11 月 11 11:26 config-2.6.32-431.el6.i686
drwxr-xr-x. 3 root root 1024 3 月 5 03:02 efi
```

`ls -l` 命令产生的效果等同于 `ll` 命令，`ll` 是该命令用法别名。`ll` 命令列出的文件属性包含 7 个字段，分别是：文件类型及文件权限、连接数、拥有者、所属组、文件大小、文件最近修改时间、文件名。

**示例：**在 `Bash` 命令中使用通配符 “\*” 来匹配目录或文件名的引用。

```
[demo@365linux ~]$ ls /etc/a*.conf
/etc/asound.conf /etc/autofs_ldap_auth.conf
```

### 3) 查找系统文件

**示例：**使用 `find` 命令查找系统文件。

```
[demo@365linux ~]$ find /home/demo -name *bash*
/home/demo/.bash_history
/home/demo/.bash_logout
/home/demo/.bashrc
/home/demo/.bash_profile
```

### 4) 文件的基本操作

**示例：**对于目录和文件的基本操作命令。

**Step 01** 在用户的家目录创建一个新的文件夹 `test03`。

```
[demo@365linux ~]$ mkdir test03
```

**Step 02** 进入刚刚创建的新目录 `test03`，创建一个空文件 `hello.txt`。

```
[demo@365linux ~]$ cd test03/
[demo@365linux test03]$ touch hello.txt
```

**Step 03** 查看文件的类型。

```
[demo@365linux test03]$ ll hello.txt
-rw-rw-r--. 1 demo demo 0 3 月 17 18:13 hello.txt
```

**Step 04** 建立 hello.txt 的软链接文件。

```
[demo@365linux test03]$ ln -s hello.txt ln_hello.txt
[demo@365linux test03]$ ll
总用量 0
-rw-rw-r--. 1 demo demo 0 3月 18 08:34 hello.txt
lrwxrwxrwx. 1 demo demo 9 3月 18 08:35 ln_hello.txt -> hello.txt
```

在 Linux 系统中，软链接文件即指向目标文件的快捷方式，但源文件被删除时，软链接则成为一个失效的文件。除了软链接文件，Linux 系统还支持硬链接文件（同样使用 `ln` 命令创建，不使用 `-s` 选项）。

**Step 05** 删除该文件和链接文件。

```
[demo@365linux test03]$ rm hello.txt
[demo@365linux test03]$ ll
总用量 0
lrwxrwxrwx. 1 demo demo 9 3月 18 08:35 ln_hello.txt -> hello.txt
[demo@365linux test03]$ rm ln_hello.txt
[demo@365linux test03]$ ls
```

**Step 06** 复制一个文件到当前工作目录。

```
[demo@365linux test03]$ cp /etc/man.config ./
[demo@365linux test03]$ ls
man.config
```

**Step 07** 创建一个新的目录 test04，并将当前目录中的 man.config 移动到 test04 目录中。

```
[demo@365linux test03]$ mkdir test04
[demo@365linux test03]$ mv man.config test04/
[demo@365linux test03]$ ls
test04
[demo@365linux test03]$ ls test04/
man.config
```

**Step 08** 进入 test04 目录，查看当前绝对路径。

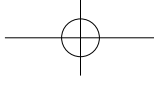
```
[demo@365linux test03]$ cd test04/
[demo@365linux test04]$ pwd
/home/demo/test03/test04
```

**Step 09** 返回上一级目录，删除含有文件的目录 test04。

```
[demo@365linux test04]$ cd ..
[demo@365linux test03]$ rm -rf test04
[demo@365linux test03]$ ls
```

**Step 10** 回到上一级目录，并删除空目录 test03。

```
[demo@365linux test03]$ cd ..
[demo@365linux ~]$ rmdir test03/
```

**REAL-WORK TIP**

`rm -rf` 命令会强制删除它的目标目录下的所有内容，所以要谨慎使用，特别是 root 用户使用时。

**NOTE**

关于 Linux 文件系统、文件类型、权限等概念会在后面的章节中提及

## 2. 设置系统时间和时区

示例：使用 `date` 命令查看当前的时间。

```
[demo@365linux test03]$ date
2014 年 03 月 18 日 星期二 08:42:51 CST
```

示例：使用 `date` 命令设置时间。

```
[root@365linux ~]# date -s "2014/03/18 08:45:00"
2014 年 03 月 18 日 星期二 08:45:00 CST
```

示例：修改系统的时区。

```
[root@365linux ~]# ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

另外，建议同时修改配置文件 `/etc/sysconfig/clock` 的内容为 `ZONE="Asia/Shanghai"`。

**REAL-WORK TIP**

使用 `date` 命令查看和修改的是系统时间。而在系统内部还有另外一个时间概念：硬件时钟，Linux 系统重启后时间会与硬件时钟保持同步，所以修改系统时间，同时也要修改硬件时钟。在命令行中使用 `hwclock` 命令查看和设置硬件时钟

## 3. 退出、关闭系统

示例：使用 `poweroff` 命令关闭 Linux 系统。

```
[root@365linux ~]# poweroff
```

在 Linux 系统中，有多种方式实现关机或重启。可以使用 `poweroff`、`shutdown`、`halt`、`init` 命令关机。使用 `shutdown`、`halt`、`init`、`reboot` 命令可以实现系统重启。关机命令之间存在互相调用，且对于关闭系统和关闭电源在不同的系统版本上存在差异。一般情况下，建议使用 `poweroff` 命令关机，使用 `reboot` 命令重启。

## 4.8 Bash命令行重要的高阶功能

### 1. 管道

在 Linux 系统中，管道允许用户将标准输出信息从程序连接至一个程序的输入，这样可



以将多个程序（命令）连接成一个管道，后一个程序的作用对象即为前一个程序的输出结果。

RHEL 系统使用竖线“|”连接程序管道操作。

**示例：**ls 命令列出 /usr/lib 目录下的所有文件，其结果并不直接显示到屏幕输出，而是通过管道发送到下一个命令 grep，通过 grep 命令过滤出结果中文件名含有 jpeg 的文件。

```
[demo@365linux ~]$ ls /usr/lib/ |grep "jpeg"
```

**示例：**用 find 命令找到 /var 目录中大小超过 1MB 的文件，将结果用 grep 过滤出文件路径中包含 cache 的文件，再将文件列出结果交给 wc 统计行数，最终得到 /var/ 目录中大小超过 1MB 且与 cache 有关的文件有多少个。

```
[root@365linux ~]# find /var -size +1M |grep "cache" |wc -l
```

## 2. I/O 重定向

Linux 命令行 I/O 重定向允许用户将标准输出或错误输出从程序发送至文件，以进行保存或屏蔽在终端的输出显示。重定向还支持反过来将文件内容读取至命令程序。关于 I/O 重定向的定义如表 4-4 所示。

表4-4 标准输入/输出

名 称	说 明	编 号	默 认
STDIN	标准输入	0	键盘
STDOUT	标准输出	1	终端
STDERR	标准错误	2	终端

**示例：**将 date 命令的标准输出重定向到文件，操作会覆盖文件原来的内容。

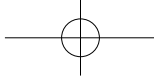
```
[demo@365linux ~]$ date > file
[demo@365linux ~]$ cat file
2014 年 03 月 19 日 星期三 12:21:43 CST
[demo@365linux ~]$ date > file
[demo@365linux ~]$ cat file
2014 年 03 月 19 日 星期三 12:21:56 CST
```

**示例：**使用重定向功能合并文件。

```
[demo@365linux ~]$ echo "1" > file1
[demo@365linux ~]$ echo "2" > file2
[demo@365linux ~]$ cat file1 file2 > file3
[demo@365linux ~]$ cat file3
1
2
```

**示例：**使用追加模式（不覆盖）将标准输出重定向到文件。

```
[demo@365linux ~]$ date >> test.txt
[demo@365linux ~]$ date >> test.txt
[demo@365linux ~]$ ls >> test.txt
[demo@365linux ~]$ cat test.txt
```



```
2014 年 03 月 19 日 星期三 12:27:00 CST
```

```
2014 年 03 月 19 日 星期三 12:27:07 CST
```

```
a
```

```
b
```

```
c
```

```
...
```

**示例：**将标准错误输出重定向到文件（覆盖）。

```
[demo@365linux ~]$ ls /boot /root 2> file
```

**示例：**将标准错误输出重定向到文件（追加）。

```
[demo@365linux ~]$ ls /boot /root 2>> file
```

**示例：**将标准错误输出重定向到设备文件 `/dev/null`，作用是丢弃错误。

```
[demo@365linux ~]$ ls /xyzabc 2> /dev/null
```

**示例：**将标准输出和标准错误输出组合，重定向到文件。

```
[demo@365linux ~]$ ls /boot /xyzabc >file 2>&1
```

或者：

```
[demo@365linux ~]$ ls /boot /xyzabc &> file
```

**示例：**使用标准输入作为 `cat` 命令的处理对象（效果等同于使用参数方式）。

```
[demo@365linux ~]$ cat < file
```

**示例：**将文件 `file` 的内容读取出来，重定向到 `test` 文件。

```
[demo@365linux ~]$ cat > test < file
```

**示例：**将键盘输入的内容重定向到 `file03` 文件，直到用户输入 EOF 结束输入。

```
[demo@365linux ~]$ cat > file03 <& EOF
```

```
> 1234
```

```
> abcd
```

```
> EOF
```

```
[demo@365linux ~]$ cat file03
```

```
1234
```

```
abcd
```

### 3. 正则表达式

正则表达式是用来搜索和匹配文本模式的特殊字符串，依赖于特定的 Linux 命令行工具工作，比如 `less`、`man`、`vim`、`locate`、`grep`、`sed`、`awk` 等，多用于脚本中批量处理。

**示例：**匹配文件中以 `b` 开头的行，这里的“`^`”是正则表达式，表示行首。

```
[demo@365linux ~]$ grep '^b' /etc/passwd
```



#### NOTE

正则表达式将在 Shell 脚本编程中详细介绍。





## 4.9 编写Bash脚本

Linux Bash 除了能够在终端中进行交互执行命令外，还可以通过编写 Shell 脚本完成自动化、批处理的任务。Shell 脚本实际上就是按语句序列执行的命令组合的文本文件，当然，在 Bash 提供了很多的编程结构，比如判断、循环、函数等。

**示例：**查看本网段哪些 IP 可以 ping 通。

```
[demo@365linux ~]$ vim ping.sh
#!/bin/bash
#Name : ping.sh
#Last revision date : 2014-01-07
#usage : ./ping.sh
#Environment
Net="192.168.1"
#script start
for i in {1..254}
do
    (
        ping -c 1 $Net.$i &>/dev/null
        if [ $? -eq 0 ];then
            echo "$i is up."
        fi
    )&
done
#wait
```



### NOTE

关于 Shell 脚本编程将在本教程的高级部分中涉及。



## 本章总结

Linux 具有强大的命令行功能，命令行具有简洁高效、功能全面、传输数据量小等特点，在系统管理中发挥巨大的作用。即使操作系统图形界面日益成熟，对于服务器系统而言，命令行功能一直被保留下来，并得到加强。

对于 Linux 命令行的学习，不能死记硬背命令的语法或用法示例，而要掌握 Linux 命令的书写规范，理解每条命令背后的逻辑、原理和命令执行后产生的效果。至于命令的用法，则要通过帮助文档、手册获得其相关的选项、参数的意义。对于常用的命令，则能熟能生巧、举一反三。

对于初学者而言，命令的使用往往成为入门的门槛，觉得生涩难懂而又难以记忆。其实



不然，对于命令的学习，不要跳脱应用而单一地对于命令选项、参数、用法去练习、记忆，而应该从实用的角度，当在具体的系统管理操作时，引入命令的学习。比如，我们知道了在图形界面下的文件管理、时间配置、退出登录关闭系统，那么在命令行下如何控制实现这些操作呢？比如，以后在用户管理中，使用命令行如何添加、删除用户等？这样就自然而然完成了对于命令的学习。学习 Linux 系统管理不是学习命令，命令只是基本单元，只是工具，利用系统命令完成系统管理的任务才是目的。

限于篇幅和保持文档的难易度，本章节中命令涉及的 Linux 系统体系概念并未展开详细介绍，需视情况由讲师教授，学生笔记，或另开篇幅详解。

### 章节重点：

- 理解命令行的格式及用法。
- 熟练使用 `--help` 和 `man` 获得命令的帮助信息。
- 理解命令行中工作路径的意义和使用。



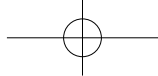
## 基础技能演练

### 1. 在命令行中完成以下操作：

- (1) 在虚拟终端前台打开 `gedit` 文本编辑器，然后将其放到后台运行。
- (2) 使用 `ls` 命令列出 `/boot` 目录下的文件，按文件的大小从小到大排列，大小显示出单位。
- (3) 使用 `date` 命令修改系统的时间，并使用 `hwclock` 命令同步硬件时钟。
- (4) 使用 `find` 命令查找系统中名字中包含 `yum` 的目录。
- (5) 查看系统中 `z` 开头的命令有哪些。
- (6) 使用 `cat` 命令查看 `/etc/fstab` 文件内容，并通过 `man` 手册理解该文件每个字段的含义。
- (7) 演示在命令行中“删除整行”、“调出历史参数”、“光标首尾移动”快捷键。
- (8) 使用 `head` 命令查看 `/etc/passwd` 文件的前 10 行。
- (9) 使用 `tail` 命令查看 `/etc/passwd` 文件的后 5 行。
- (10) 使用 `less` 命令查看 `/etc/man.config` 文件内容，并进行翻页操作，搜索 `man` 关键字。

### 2. 使用命令行对目录和文件进行以下操作：

- (1) 使用 `touch` 命令在家目录创建一个文件 `test.txt`。
- (2) 使用 `mkdir` 命令在家目录创建一个目录 `test`。
- (3) 使用 `cp` 命令复制 `test.txt` 文件到 `test` 目录中，并改名为 `test.txt.backup`。
- (4) 使用 `rm` 命令删除家目录下的 `test.txt` 文件。
- (5) 使用 `mv` 命令将 `test` 目录中的 `test.txt.backup` 文件改名为 `test.txt`。
- (6) 使用 `cp` 命令将 `/usr/share/doc` 目录下 `dhc` 开头（提示：使用通配符 `*`）的目录和文件



全部复制到家目录下的 `test` 目录。

(7) 使用 `mv` 命令从 `test` 目录中移动 `test.txt` 到家目录；使用 `rm` 命令删除整个 `test` 目录。

3. 最后使用 `poweroff` 命令关闭系统（重启系统的命令是 `reboot`）。

## 提高技能实践

使用命令行对文件进行打包和压缩。

(1) 在家目录下创建空目录 `xueing`。

(2) 使用 `locate` 命令找到 `linux.words` 文件的位置。

(3) 进入 `xueing` 目录，将 `linux.words` 复制到当前目录，查看文件的大小。

(4) 使用 `gzip` 命令对 `linux.words` 进行压缩，比较先后的大小。

(5) 使用 `gunzip` 命令对 `linux.words` 进行解压缩。

(6) 思考如何在压缩时保留原文件。

(7) 使用 `bzip2` 命令进行同样的压缩和解压缩的操作。

(8) 退出到上一级目录，使用 `tar` 命令对 `xueing` 目录进行打包操作（如果要同时压缩应该怎么做）。

(9) 对刚刚打包的 `xueing` 目录进行解包的操作。



## 第5章

# vim编辑器

编辑器是编写或修改文本文件的重要工具之一，在各种操作系统中，编辑器都是不可缺少的部件。在 Linux 操作系统中，系统和应用的配置大多需要通过修改配置文件来进行设置，所以编辑器的使用频率高，更显得重要。熟练掌握 Linux 编辑器的用法，可以极大地提高工作效率。为方便各种用户在各个不同的环境中使用，Linux 提供了一系列的编辑器，如 vi、emacs、pico、nano、gedit 等。每种编辑器都有各自的特性，在不同的工作环境中根据自己的需求加以选择。

vim (vi improved) 是一种强大的文本编辑器，支持复杂的文本操作。相对图形界面的 gedit 编辑器，vim 可以很方便地在命令行中使用，而且在任何 Linux 系统中始终可用。

vim 是 vi 的高级版本，提供更多的功能，比如自动格式、语法高亮等。当系统中 vim 无法使用时，依然可以使用 vi 命令代替，用法相同。



### REAL-WORK TIP

在大多数采用最小化安装的 Linux 系统上，vim 默认是不被安装的，但有 vi 可以使用，用户也可以额外安装 vim 编辑器。

## 5.1 在命令行中使用vim编辑器

在命令行终端中使用 vim 命令启动 vim 编辑器。

**示例：**使用 vim 打开 filename.txt 文件，如果文件不存在，则保存后创建一个新文件。

```
[demo@365linux ~]$ vim filename
```



## 5.2 vim的3种模式

### 1. 命令模式

打开 vim 编辑器，即进入命令模式（也称一般模式）。通过键盘命令，对文档进行复制、粘贴、删除、替换、移动光标、继续查找等。该模式也是编辑模式和末行模式切换的中间模式，可以通过 Esc 键返回到命令模式。

### 2. 编辑模式

也称插入模式，用于对文档进行添加、删除、修改等操作。在编辑模式中，所有的键盘操作（除了退出编辑模式键）都是输入或删除的操作，所以在编辑模式下没有可用的键盘命令操作。

### 3. 末行模式

进入末行模式，光标移动到屏幕的底部，输入内置的指令，可执行相关的操作，如文件的保存、退出、定位光标、查找、替换、设置行标等。

### 4. 3 种模式之间的切换

3 种模式之间的切换如图 5-1 所示。

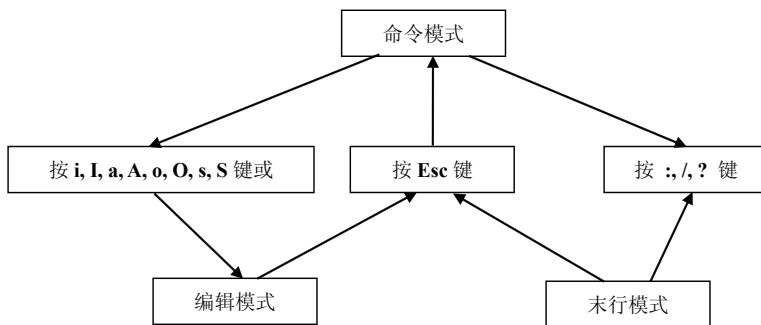


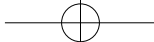
图5-1 vim 3种模式切换

3 种模式切换按键详细说明如下。

(1) 进入编辑模式的命令如表 5-1 所示。

表5-1 进入编辑模式的命令

按 键	功 能
i	从光标所在位置前面开始插入文本，同 insert 键
I	从光标所在行的行首开始插入文本
a	从光标所在位置后面开始插入文本



续表

按 键	功 能
A	从光标所在行的行尾开始插入文本
o	在光标所在行下方新增一行插入文本
O	在光标所在行上方新增一行插入文本
s	删除光标所在字符并开始插入文本
S	删除光标所在行并开始插入文本

(2) 进入末行模式的命令如表 5-2 所示。

表5-2 进入末行模式的命令

按 键	功 能
:	在后面接要执行的命令
/	在后面接要搜索的字符串，从光标位置开始向下搜索，按 n 重复前一个搜索动作，按 N 反向重复前一个搜索动作
?	在后面接要搜索的字符串，从光标位置开始向上搜索，按 n 重复前一个搜索动作，按 N 反向重复前一个搜索动作

**NOTE**

不能从编辑模式直接进入末行模式，也不能从末行模式直接进入编辑模式，二者之间的转换必须先按 Esc 键退出到命令模式，然后再通过相关的命令按键切换

## 5.3 退出vim编辑器

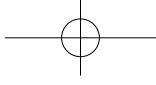
退出 vim 需要进入末后模式执行退出命令。在命令模式下，按 “:” 键进入末行模式，在末行模式下输入相关的命令，如表 5-3 所示。

表5-3 退出vim编辑器的命令

命 令	功 能
q	没有对文档做过修改，退出
q!	对文档做过修改，强制不保存退出
wq 或 x	保存退出；可以添加! 表示强制保存退出
ZZ	若文档没有修改，则不保存退出；若文档已经修改，则保存后退出

## 5.4 vim命令模式下的常用操作

在 vim 编辑器命令模式下，有着大量方便快捷的键盘命令，用来控制光标、操作文本。



使用 vim 的键盘命令，可以使用户的双手不离开主键盘区域，不使用鼠标，实现光标移动、复制、粘贴、删除等操作，熟练使用，将极大地提高工作效率。

以下列出命令模式下常用的操作命令。

## 1. 光标移动

常用的光标移动命令如表 5-4 所示。

表5-4 光标移动命令

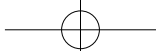
命令 (n 表示数字)	功 能
h/j/k/l	光标向左 / 下 / 上 / 右移动一个字符
nj	向下移动 <i>n</i> 行 (可以是 nh/nk/nl)
Ctrl+f/b/d/u	屏幕向下 / 上 / 移动一页 (半页)
n<Space 键>	光标向后移动 <i>n</i> 个字符
n<Enter 键>	光标向下移动 <i>n</i> 行
H/M/L	光标移动到屏幕上方 / 中央 / 下方
+/-	光标移动到非空格符的下 / 上一行
0 或者 ^	光标移动到行首
\$	光标移动到行尾
gg	光标移动到文件第一行
G	光标移动到文件最后一行
nG	光标移动到文件的第 <i>n</i> 行

## 2. 删除、复制与粘贴

常用的删除、复制与粘贴命令如表 5-5 所示。

表5-5 删除、复制与粘贴命令

命 令	功 能
x/X/nx	向后 / 前删除一 ( <i>n</i> ) 个字符
dd/ndd	删除光标所在的行 / 向下删除 <i>n</i> 行
d1G/dgg	删除光标位置到第一行的所有数据
d G	删除光标位置到最后一行的所有数据
d0/d\$	删除光标位置到该行行首 / 尾
cw/ncw	更改光标位置的一 ( <i>n</i> ) 个字符串
yy/nyy	复制光标所在一 (向下 <i>n</i> ) 行
y1G/ygg	复制光标位置到第一行所有行
yG	复制光标位置到最后一行数据
y0/y\$	复制光标位置到该行行首 / 尾
yw/nyw	复制光标位置一 ( <i>n</i> ) 个字符串
p/P	粘贴到光标位置下 / 上一行



### 3. 替换

常用的替换命令如表 5-6 所示。

表5-6 替换命令

命 令	功 能
r	仅替换一次光标所在的字符
R	一直替换光标所在字符，直到按 Esc 键

### 4. 其他

其他命令模式下常用的命令如表 5-7 所示。

表5-7 其他命令模式常用命令

命 令	功 能
u	撤销前一个操作
U	撤销一行内的所有改动
Ctrl+r	重做上一个操作
J	合并光标所在行与下一行

## 5.5 vim末行模式下的常用操作

在 vim 末行模式下，除了最常用的保存、退出等命令外，有更多的命令组合用于完成复杂的文本操作。

以下列出末行模式下常用的操作命令。

### 1. 搜索替换

搜索替换的命令如表 5-8 所示。

表5-8 搜索替换命令

命 令	功 能
n1,n2s/word1/word2/g	将从 n1 行到 n2 行之间的 word1 替换为 word2，如无 g 则只替换第一个匹配
1,\$s/word1/word2/gc	将从第一行到最后一行之间的 word1 替换为 word2，c 表示每次替换确认
%s/^/word2/g	在整个文件的每行行首插入 word2
%s/\$/word2/g	在整个文件的每行行尾插入 word2
%/var/char-&/g	在整个文件中匹配到 var 后替换为 char-var,& 指代匹配的结果，可能为正则匹配的多种结果

### 2. 其他

其他末行模式常用命令如表 5-9 所示。



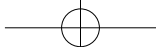


表5-9 其他末行模式常用命令

命 令	功 能
w/w!	保存文件 / 强制保存文件
w filename	另存为 filename
n1,n2 w filename	将文件的第 n1 行到第 n2 行另存到 filename
r filename	读取另外的文件到正在编辑的文件
! command	暂时离开 vi 执行命令
r !command	把命令的输出插入到当前
sh	转动 Shell, 输入 exit 返回
e!	将文件还原
set nu/set nonu	设置行号 / 取消行号
set autoindent	设置自动对齐格式 (取消 set noautoindent)
set ruler	设置在屏幕底部显示光标所在的行列位置
set ignorecase	忽略正则表达式中的大小写
nohlsearch	取消搜索到的关键字的高亮显示

**REAL-WORK TIP**

在末行模式下输入命令时可以用 Tab 键自动补全, 也可以使用 --help 命令获得相关帮助。

## 5.6 vim的其他操作

### 1. 块操作

命令模式输入 v 则进入块操作: 移动光标选定操作块; 按 y 键复制; 按 c 键剪切; 按 p 键粘贴。

### 2. 分隔窗口

可以在一个 vim 当前窗口中并排打开多个文件。

示例: 水平分隔窗口命令, 同时水平排列打开的文件 file1.txt 和 file2.txt。

```
[demo@365linux ~]$ vim -o file1.txt file2.txt
```

示例: 垂直分隔窗口命令, 同时垂直排列打开的文件 file1.txt 和 file2.txt。

```
[demo@365linux ~]$ vi m -O file1.txt file2.txt
```

窗口移动快捷键: Ctrl+w。

### 3. vim 文件恢复

使用 vim 编辑一个文件 test.txt 时, 会在文件所在目录产生一个临时文件, 文件名为 .test。



txt.swp，这是一个隐藏文件，我们在 vi 中所做的操作会暂时存放在该文件内。

如果在文件编辑过程中 vi 非正常关闭，那么重新打开 vi test.txt 时，系统会提示发现交换文件 .test.txt.swp，可能的原因是：

- 有另一个程序也在编辑同一个文件。
- 上次编辑此文件时崩溃。

这时可以按 O 只读打开，或 R 进行修复，或 E 直接编辑，或 Q 退出。

手动删除 .test.txt.swp 后，则不会再出现该提示。



### REAL-WORK TIP

在对系统的关键性配置文件进行编辑修改时，强烈建议先做好原始文件的备份，因为 vim 编辑器对文件完成修改并保存后是无法恢复的。



## 本章总结

vim 编辑器是 Linux 系统中最常用的命令行工具。看似繁多的操作命令其实有规律可循，记住几个基础指令，其他的技巧多为基础指令的组合，可以在此基础上举一反三。

章节重点：

vim 3 种模式的切换和使用。



## 基础技能演练

1. 使用 vim 创建一个文件。

- (1) 打开一个空文件（比如 vim filename.txt）。
- (2) 切换到编辑模式。
- (3) 输入文本（任意字符）。
- (4) 使用 Esc 键返回到命令模式。
- (5) 按 “:” 键进入末行模式。
- (6) 输入 “x” 命令保存退出。

2. 使用 vim 编辑文件。

- (1) 打开虚拟终端，切换到 root 用户。
- (2) 使用 vim 打开文件 /tmp/test.file.txt。
- (3) 按 i 键进入插入模式，然后输入以下文本：

```
Change the last word in this line to goose: duck.  
Quack Quack Remove the first two Quack words.
```



```
Copy this line so it apperars twice.  
Make sure this is the first line in the file.
```

- (4) 按 **Esc** 键并使用 **1G** 组合键返回到该文件的第一行。
  - (5) 在第一行，将单词 **duck** 更改为 **goose**。
  - (6) 在第二行，删除前两次出现的单词 **Quack**。
  - (7) 复制第三行，然后进行粘贴使该行出现两次。
  - (8) 删除最后一行，然后粘贴，使其显示为该文件的第一行。
- 完成之后，整个文件将显示如下：

```
Make sure this is the first line in the file.  
Change the last word in this line to goose: goose.  
Remove the first two Quack words.  
Copy this line so it apperars twice.  
Copy this line so it apperars twice.
```

- (9) 保存并退出文件（输入 **:wq**）。

### 3. 使用 vim 编辑系统的配置文件。

- (1) 使用 **vim** 打开系统的配置文件 **/etc/issue**。
- (2) 在文件的顶部添加一行 **Welcome**，保存退出。
- (3) 通过 **Ctrl+Alt+F2** 组合键切换到虚拟控制台，并按 **Enter** 键刷新提示，测试更改过的效果，文本 **Welcome** 将会显示在登录提示符的顶部。



## 提高技能实践

按照步骤完成以下任务：

- (1) 请在 **/tmp** 目录下建立一个名为 **vitest** 的目录，进入 **vitest** 目录中。
- (2) 将 **/etc/man.config** 复制到本目录中，并重命名为 **man.config.back**。
- (3) 使用 **vi** 编辑器打开本目录下的 **man.config.back**。
- (4) 在 **vi** 中设置行号。
- (5) 光标移动到第 58 行，向右移动 40 个字符，记下您的光标所处位置。
- (6) 移动到第一行，并且向下搜索“**bzip2**”字符串，请问它在第几行？
- (7) 接下来，要将第 50 ~ 100 行之间的 **man** 改为 **MAN**，要求提示确认。
- (8) 修改完后，突然反悔了，要全部复原，有哪些方法？
- (9) 要复制第 51 ~ 60 行的内容，并粘贴到最后一行之后。
- (10) 删除第 11 ~ 30 行之间的 20 行。
- (11) 将这个文件另存成一个 **man.test.config**。
- (12) 到第 29 行，并且删除 15 个字符。
- (13) 存储后离开。



## 第6章

# 管理用户和组

Linux 系统是一个真正意义上的多用户多任务的系统，这就意味着在系统的机制上存在着多种不同的用户，对应着不同的权限和功能。系统中的用户可以是一个对应真实物理用户的账号，也可以是特定应用程序使用的身份账号。Linux 系统通过定义不同的用户，来控制用户在系统中的权限。系统的每个文件都被设计成属于相应的用户和组，不同的用户则决定了其对系统内哪些文件是否可以访问、写入或执行。

在 RHEL 中，有 3 种不同的文件类型：**root** 用户（也称管理员账户、超级用户或根用户）、普通用户和系统用户。

**root** 用户是系统内置的管理员用户，拥有系统最大权限，可以完全访问和操作系统的所有文件。正因如此，在通常系统使用时，不建议使用 **root** 用户登录系统，只有在需要管理员权限时才由普通用户切换到管理员。

如果在安装系统时安装了图形界面，那么在第一次启动的时候会被要求创建一个普通用户，这个用户是被用来登录和使用系统的。普通用户默认情况下只能对自己的家目录下的文件和文件夹进行修改或删除的操作，最大限度地控制了对系统的损害。如果在安装系统时没有安装图形环境，也应在进入系统后手动创建一个普通用户，用作平常登录使用。

系统用户类似于普通用户，不同之处在于系统用户通常没有自己的家目录，也不能登录到系统，仅用来控制应用程序的运行。系统用户一般由系统自带应用或其他服务程序的软件包安装时创建。

在 RHEL 的设计中，每个用户都有一个对应的组，组即是多个（含一个）成员用户为同一目的组成的组织，组内的成员对属于该组下的文件拥有相同的权限。默认情况下，RHEL 用户拥有自己的私人组（**U**sr **P**riate **G**roup，**UPG**），当一个新用户被创建时，同时会创建一个和用户名相同的用户私人组。

## 6.1 使用图形环境用户管理工具

有两种方式对用户和组进行管理：使用图形工具和使用命令行。

用户图形配置工具可以进行查看、修改、创建、删除本地用户和组的操作。

在 GNOME 桌面菜单中，依次单击“系统”→“管理”→“用户和组群”，在弹出的对话框中输入管理员的密码完成授权，即打开“用户管理者”配置窗口，如图 6-1 所示。

或者在命令行终端中输入命令 `system-config-users` 打开。

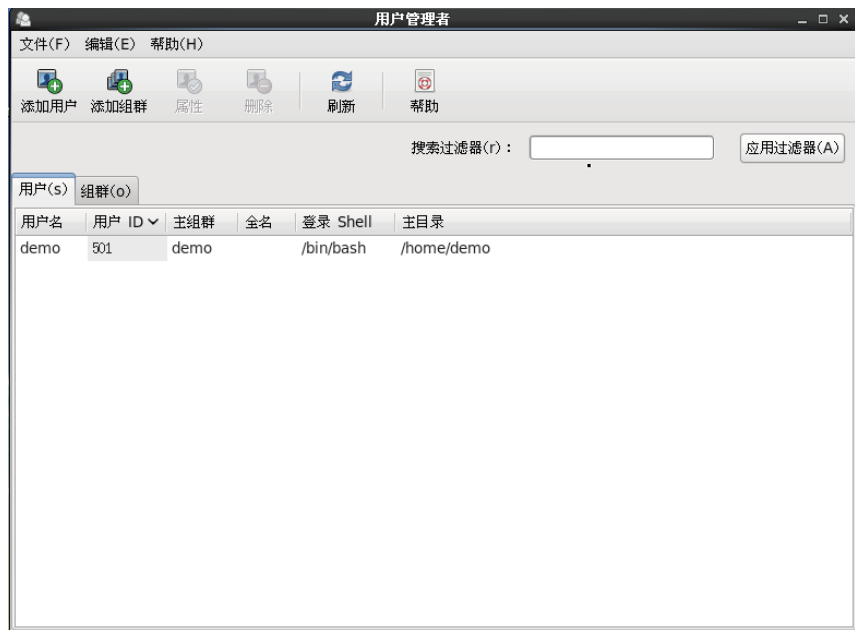


图6-1 用户管理配置工具

### 1. 查看用户和组

在主窗口中有两个标签：用户和组群。在用户的标签中列出了系统的普通用户和用户的相关属性信息，包括用户名、用户 ID、主组群、全名、登录 Shell、主目录。在组群标签中列出了系统本地用户的组和组属性信息，包括组群名、组群 ID、组群成员。

如果用户较多时，可以使用搜索过滤器进行搜索。

RHEL 系统中普通用户和组的 ID 默认从 500 开始，0 ~ 499 为系统用户保留。在“用户管理工具”中不显示系统用户。如果要查看系统用户，可以在单击“编辑”→“首选项”，在弹出的对话框中取消勾选“隐藏系统用户和组”复选框。

### 2. 添加新用户

要添加新用户，则单击“添加用户”按钮。

在“添加新用户”对话框中填入用户信息（用户名、全称、密码），示例如图 6-2 所示。



图6-2 添加新用户

**NOTE**

密码的设置至少 6 位，为安全起见，RHEL 系统不允许使用字典密码。如果密码过于简单，系统会提示密码不建议使用。建议密码包含字母、数字和特殊字符的组合。

登录 Shell 的下拉菜单允许为用户指定一个登录 Shell。默认使用 `/bin/bash`，这也是通用配置。您可以为特殊应用的用户选择不同的 Shell，比如选择 `/sbin/nologin` 可以禁止用户登录。

默认情况下，新建普通用户会同时创建用户的家目录，位于 `/home/username`。您可以取消创建家目录或者更改用户家目录的路径。在创建用户的家目录时，系统会自动复制 `/etc/skel/` 目录下的默认配置文件到新用户的家目录。

创建用户时系统会自动创建一个和用户名相同的私人组作为新用户的默认所属主组。您可以取消创建用户私人组，而为其指定一个已存在的组。

如果不手动指定用户的 ID，系统则自动分配一个大于 500 且未使用的 ID 给新用户。不建议给普通用户手动指定一个小于 500 的 ID。

### 3. 添加新组

单击“添加组群”按钮打开“添加新组群”对话框，如图 6-3 所示。在对话框中输入新组的组群名，也可以手动指定组群 ID。如果不指定的话，系统同样自动分配一个大于 500 的 GID 给当前的新建组群。



图6-3 添加新组群

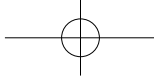
#### 4. 修改用户属性

要查看或修改用户属性，单击选择要查看的用户，然后单击“属性”按钮，则弹出“用户属性”对话框，如图 6-4 所示。



图6-4 用户属性

- 用户数据：基本的用户配置信息，包括用户名、全称、密码、主目录及登录 Shell。
- 帐号信息：可以设置用户账号的过期日期；可以锁定用户密码，限制用户登录。
- 密码信息：设置密码过期时间。可设置的期限有如下几种。
  - 允许更换前的天数：密码最少使用时间，即密码必须要使用多少天后才能被修改。
  - 需要更换前的天数：密码最长使用时间，即密码使用多少天后必须被修改。
  - 更换前的警告天数：密码即将过期前的警告天数。
  - 账号被取消激活前的天数：密码过期几天后账号失效，在此段时间内要求用户修改密码。
  - 在该选项卡内还可勾选“下次登录强制修改密码”复选框。
- 组群：可以将用户加入多个组群，在主组群的下拉选项中可选择修改用户的主组群。



## 5. 修改组属性

要查看或修改组的属性，同样是在组列表选定相应的组，然后单击“属性”按钮，弹出“组群属性”对话框，如图 6-5 所示。组群可设置的属性比较简单，主要是可以在组群用户的页面中添加或删除组群的成员。

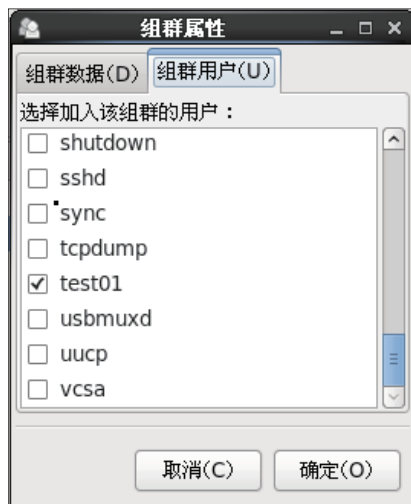


图6-5 组群属性

## 6. 删除用户和组

选定要删除的用户或组，单击“删除”按钮。在删除用户时系统会询问是否同时删除用户的文件，包括家目录、邮件目录及临时文件。如果用户的私人组中没有别的成员，则删除用户的同时会删除该用户的私人组。

# 6.2 使用命令行工具管理用户

除了可以使用图形用户管理工具来配置用户，RHEL 同样提供了一系列的命令行工具来管理用户，可以实现添加、修改、删除用户，甚至更多。

### 1. 使用 useradd 命令创建用户

用法：useradd [options] user\_name

常用选项：

- -u UID：为新用户指定一个 UID（不使用系统默认按顺序分配的），使用 -r 强制建立系统账号（小于 /etc/login.defs 上 UID\_MIN），使用 -o 允许新用户使用不唯一的 UID。
- -g GROUP：为新用户指定一个组（指定的组必须存在）。





- **-G GROUPS** : 为新用户指定一个附加组。
- **-M** : 不创建用户的家目录 (默认创建)。
- **-m** : 为新用户创建家目录。使用 **-k** 选项将 `skeleton_dir` 内的档案复制到家目录下。
- **-c** : 为新用户进行说明注释 (`/etc/passwd` 的说明栏)。
- **-d** : 为新用户指定家目录。默认值为 `default_home` 内的 `login` 名称。
- **-s** : 为新用户指定登录后使用的 `Shell`。
- **-e** : 为新用户指定账号的终止日期。日期的指定格式为 `MM/DD/YY`。
- **-f** : 用户账号过期几日后永久失效。当值为 `0` 时账号则立刻失效, 为 `-1` 时关闭此功能。默认关闭。

**示例** : 使用命令创建新用户。

```
[demo@365linux ~]$ su -
密码:
[root@365linux ~]# useradd user01
```

创建用户的操作需要管理员权限, 如果您是以普通用户登录系统, 则首先要切换到管理员用户。

**示例** : 创建新用户, 使用自定义选项。

```
[root@365linux ~]# useradd -u 1000 -c "ftp user" -s /sbin/nologin user02
```



#### REAL-WORK TIP

作为一个好的习惯, 您应该给每个账号添加注释说明 (使用 **-c**), 否则时间长了, 您可能会忘记每个用户的用途

## 2. 使用 `passwd` 命令设置用户密码

用法 : `passwd [options] user_name`

常用选项 :

- **-l** : 锁定指定的账号。
- **-u** : 解锁指定的被锁定的账号。
- **-n** : 指定密码最短时间。
- **-x** : 指定密码最长时间。
- **-w** : 指定密码过期前的警告天数。
- **-i** : 指定密码过期后, 账号失效前的天数。
- **-S** : 报告指定用户密码的状态。
- **--stdin** 从标准输入读入密码, 常用于 `Shell` 脚本。

**示例** : 为用户设置密码。

```
[root@365linux ~]# passwd user01
更改用户 user01 的密码。
```



新的 密码：  
重新输入新的 密码：  
passwd：所有的身份验证令牌已经成功更新。

**NOTE**

root 用户可以修改任何用户的密码，普通用户仅可以修改自己的密码

### 3. 系统添加用户过程解释

使用 `useradd` 命令创建新用户和使用 `passwd` 命令设置用户密码时，系统是如何保存用户信息的？当命令执行时，系统发生了如下动作。

(1) 在系统用户信息配置文件中添加一行；可以使用 `cat` 命令查看。

```
[root@365linux ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
user01:x:502:502::/home/user01:/bin/bash
```

在 `/etc/passwd` 文件中，每一行对应一个用户的属性信息。新创建的用户被追加在最后一行。每一行用 “:” 分隔成 7 个字段，每个字段的意义如下。

- Account：用户名。
- Password：用户密码，因为 `/etc/passwd` 所有人可读，出于安全考虑，该字段默认用 `x` 代替。真正的密码文件保存在 `/etc/shadow` 文件中。用户可以通过 `authconfig` 来设定是否使用 `shadow` 文件及 `md5` 加密。
- UID：用户 ID。UID 为 0，表示是拥有最高权限的系统管理员（如 `root` 用户）。默认 UID 1 ~ 499 为系统保留账号，普通用户 UID 一般为 500 ~ 60 000。一般来说，用户 UID 是唯一的。
- GID：用户所属主组的组 ID。
- GECOS：用户的注释说明（可选）。
- Directory：用户的主目录。
- Shell：用户所使用的 Shell。

`useradd` 命令不使用任何选项，添加的新用户属性使用系统预设的默认值。新添加的 `user01` 用户 UID 和 GID 都是 502，家目录是 `/home/user01`，登录 Shell 为 `/bin/bash`。

(2) 在系统用户密码存储文件 `/etc/shadow` 中添加用户密码信息新行。

```
[root@365linux ~]# cat /etc/shadow
...
user01:$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4
XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::
```

`/etc/shadow` 文件用于存放用户密码相关的信息。该文件的权限值为 000（任何人没有任



何权限)，只有 **root** 用户可以突破这一限制。该文件同样用 “:” 将文件分隔成 9 个字段，每个字段的意义如下。

- 登录名。
- 经 md5 加密的用户密码（前面有 \* 或！，则账号被锁定无法使用）。
- 密码上一次被更改的日期（格式：从 1970 年 1 月 1 日起的天数）。
- 密码不可更改的天数（密码最短时间，密码要过多少天才能被修改）。
- 密码过期时间（密码最长时间，密码过多少天后必须被更改）。
- 密码过期前警告时间。
- 密码过期几天后账号失效（在此时间段内要求用户修改密码）。
- 账号失效日期（格式：从 1970 年 1 月 1 日起的天数）。
- 保留，目前未定义。

默认情况下，用户被设置为永不过期。

(3) 在组账号信息配置文件 **/etc/group** 中添加相关私人组新行。

```
user01:x:502:
```

组账号信息配置文件的 3 个字段分别表示：组名；密码标识（加密密码存储在 **/etc/gshadow** 中）；GID，GID 应该与 **/etc/passwd** 文件中对应用户的 GID 一致。

(4) 在组的密码存储文件 **/etc/gshadow** 中添加新行。

```
user01:!::
```

(5) 在系统 **/home** 目录下创建用户的同名家目录。

```
[root@365linux ~]# ls -l /home
drwx-----. 4 user01 user01 4096 3月 24 17:42 user01
```

用户家目录（也称主目录或主文件夹）属于 **user01** 用户和 **user01** 组，只有 **user01** 用户对该文件夹有读、写、执行的权限，所有其他的权限被拒绝。关于文件权限将在下一节详细说明。

(6) 将 **/etc/skel** 目录下默认的用户配置文件复制到用户的家目录。

```
[root@365linux ~]# ls -la /home/user01
总用量 28
drwx-----. 4 user01 user01 4096 3月 24 17:42 .
drwxr-xr-x. 5 root    root    4096 3月 24 18:01 ..
-rw-r--r--. 1 user01 user01   18 7月 9 2013 .bash_logout
-rw-r--r--. 1 user01 user01  176 7月 9 2013 .bash_profile
-rw-r--r--. 1 user01 user01  124 7月 9 2013 .bashrc
drwxr-xr-x. 2 user01 user01 4096 7月 14 2010 .gnome2
drwxr-xr-x. 4 user01 user01 4096 3月 5 02:54 .mozilla
```

(7) 在系统邮件存储目录创建用户邮箱文件（可选）。

```
[root@365linux ~]# ll /var/mail/
-rw-rw----. 1 user01 mail    0 3月 24 17:42 user01
```



#### 4. 使用 usermod 命令修改用户账号信息

用法：usermod [option] user\_name

常用选项：

- -L：锁定账号（在 /etc/shadow 中密码部分前加一个！），-U 用于解锁。
- -l：改变用户的登录名。

除以上选项，usermod 使用与 useradd 相似的选项和参数指定修改用户账号属性信息。

示例：修改用户账号信息的注释说明。

```
[root@365linux ~]# usermod -c "web user" user01
[root@365linux ~]# cat /etc/passwd |grep user01
user01:x:502:502:web user:/home/user01:/bin/bash
```

示例：使用 usermod 锁定和解锁账号。

**Step 01** 使用 -L 选项锁定用户，限制用户登录。

```
[root@365linux ~]# usermod -L user01
```

**Step 02** 查看用户的密码文件，此时在密码字段前添加了“!”号，用户不能登录。

```
[root@365linux ~]# grep user01 /etc/shadow
user01:!!$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4
4XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::
```

**Step 03** 使用 -U 选项解锁被锁定的用户。

```
[root@365linux ~]# usermod -U user01
[root@365linux ~]# grep user01 /etc/shadow
user01:$6$0yEadFvM$FBkTS2Tog3U9EM6Ga5Y3BdneOBTEnbGG7oGB2qVa9dpo8EyHcN.LHj4
4XFQ8zluNqeBLaUghzJlRJcmIX6lAjz1:16153:0:99999:7:::
```

#### 5. 使用 userdel 命令删除用户

用法：userdel [options] user\_name

常用选项：

- -f：强制删除用户，即使该用户仍在登录。
- -r：删除用户的同时删除该用户的家目录和邮件。

示例：删除一个用户。

```
[root@365linux ~]# userdel user01
[root@365linux ~]# ls /home
demo user01 user02
```

没有使用 -r 选项，删除了用户，而用户的家目录被保留。



#### NOTE

如果用户同名组没有其他成员，则连同删除；反之保留。

**REAL-WORK TIP**

在系统中新建一个用户时，系统可能将一个已经删除的旧用户的 UID 重新分配给新用户，如果在删除旧用户时没有删除该用户的文件（在家目录或散落在其他位置），新用户就获得旧用户原来在系统中的文件的所有权。这种情况将导致信息泄露和其他安全问题。解决的方案是，使用 `find` 命令找出这些文件进行删除或备份（改变权限）。

## 6. 使用 `groupadd` 命令新建组群

用法：`groupadd [options] group_name`

常用选项：

`-g`：指定新建组的 GID。使用 `-o` 选项，可以使用重复的 GID。使用 `-r` 选项来建立系统账号。

示例：创建一个新组。

```
[root@365linux ~]# groupadd sales
```

## 7. 组群管理命令示例

示例：给组群添加密码，使知道密码的用户能临时加入到该组。

**Step 01** 修改组 `sales` 的密码。没有设密码的组是不允许用户申请加入的。

```
[root@365linux ~]# gpasswd sales
```

**Step 02** 添加用户 `zhangsan`，切换到用户，查看用户当前所属的组群。

```
[root@365linux ~]# useradd zhangsan
[root@365linux ~]# su - zhangsan
[zhangsan@365linux ~]$ groups
zhangsan
```

**Step 03** 用户主动申请加入组群 `sales`。

```
[zhangsan@365linux ~]$ newgrp sales
密码：
[zhangsan@365linux ~]$ groups
sales zhangsan
```

**Step 04** 用户退出组群 `sales`。

```
[zhangsan@365linux ~]$ exit
[zhangsan@365linux ~]$ groups
zhangsan
```

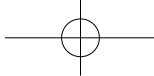
**Step 05** 删除组 `sales` 的密码，禁止以后用户再主动加入。

```
[root@365linux ~]$ gpasswd -r sales
```

示例：设置组群管理员，用来管理组成员。

**Step 01** 添加用户 `lisi`，将用户 `zhangsan` 加入到 `sales` 组中。

```
[root@365linux ~]# useradd lisi
[root@365linux ~]# gpasswd -a lisi sales
```



**Step 02** 将用户 lisi 设置为组 sales 的管理员。

```
[root@365linux ~]# gpasswd -A lisi sales
```

**Step 03** 切换到用户 lisi，行使管理权限，将用户 zhangsan 加入组 sales。

```
[root@365linux ~]# su - lisi
[lisi@365linux ~]$ gpasswd -a zhangsan sales
Adding user zhangsan to group sales
```

**Step 04** 此时用户 zhangsan 成为组 sales 的成员，切换组时不需要组密码。

```
[zhangsan@365linux ~]$ groups
zhangsan sales
[zhangsan@365linux ~]$ newgrp sales
[zhangsan@365linux ~]$ groups
sales zhangsan
```

**Step 05** 组管理员 lisi 将用户 zhangsan 从 sales 组中删除。

```
[lisi@365linux ~]$ gpasswd -d zhangsan sales
Removing user zhangsan from group sales
```

示例：修改组名，将组 sales 的组名称改为 xiaoshou。

```
[root@365linux ~]# groupmod -n sales xiaoshou
```

示例：删除组 xiaoshou。

```
[root@365linux ~]# groupdel xiaoshou
```

## 8. 其他有用的命令

### 1) 使用 id 命令输出用户简要信息

示例：查看用户 user02 的 UID、GID、组。

```
[root@365linux ~]# id user02
uid=1000(user02) gid=1000(user02) 组=1000(user02)
```

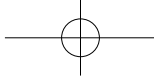
### 2) 使用 chage 命令查看或修改用户的账号和密码信息

示例：查看用户的账号和密码信息。

```
[root@365linux ~]# chage -l user02
Last password change           : Mar 24, 2014
Password expires                : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

示例：设置用户 user02 账号过期时间是 2014-12-28。

```
[root@365linux ~]# chage -E "2014-12-28" user02
```

**REAL-WORK TIP**

在创建用户时或之后为用户设置有效期,对于一些临时雇用人员开启的账号是非常有用的。账号在临时人员离开后自动失效,可以避免一些安全上的隐患。

### 3) 使用 pwck 命令对用户进行一致性检查

**示例:** 检查系统内用户账号信息的完整性。

```
[root@365linux ~]# pwck
user 'adm': directory '/var/adm' does not exist
...
pwck: 无改变
```

输出中系统账号家目录信息的缺失是正常的。

## 6.3 切换账号

在命令行中,可以很方便地切换到不同的用户。普通用户要想切换到任意别的用户,必须输入目标用户的密码;管理员可以随时切换到其他的用户而不需要用户密码。切换用户可以使用以下命令。

- **su**: 切换到其他的用户,即以目标用户的身份登录系统继续工作。
- **sudo**: 并不真的切换用户,而是以管理员的身份执行命令。

### 1. 使用 su 命令切换账号

**示例:** 用户 user02 切换到 user03,前提是 user03 设置了密码可以登录。

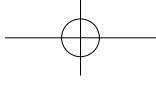
```
[user02@365linux ~]$ su - user03
密码:
[user03@365linux ~]$
```

**示例:** 用户 user03 退出登录,返回 user02。

```
[user03@365linux ~]$ exit
logout
[user02@365linux ~]$
```

**示例:** 用户 user02 切换到管理员用户 root,用户名可以省略。

```
[user02@365linux ~]$ su -
密码:
[root@365linux ~]#
```

**REAL-WORK TIP**

在切换用户时使用 `su` 命令，在 `su` 和用户名之间的选项使用 “-” 或不使用，结果存在差异。不使用时，可切换用户但不会切换到目标用户的目录和 Shell 环境变量；而使用时任何环境因素都会初始化，效果相当于目标用户登录后。

## 2. 使用 `sudo` 命令获得 root 权限

在生产环境中，特别是多用户进行项目协作的场景下，需要管理员权限时，管理员不能直接告诉用户 root 账户的密码，而应该使用 `sudo` 方式授予普通用户部分管理员权限。这在权限控制和问责机制中非常重要。

在 RHEL 系统中，普通用户要使用 `sudo` 方式临时获得管理员的权限执行命令时，需要管理员提前将这个普通用户添加到 `sudo` 配置文件的授权用户列表中。因此，在默认情况下，所有的普通用户均不可以使用 `sudo`。

**示例：**授权普通用户可以使用 `sudo` 临时获得 root 权限。

**Step 01** 使用 `visudo` 命令编辑 `sudo` 配置文件 `/etc/sudoers`。

```
[root@365linux ~]# visudo
demo ALL=(ALL) ALL
```

在文件的最后加入 `demo` 用户，这一行配置的含义是允许 `demo` 用户使用 `sudo` 在任何地方运行任何命令，即授予最大权限。

**Step 02** 切换到普通用户 `demo`，演示使用 `sudo` 方式可以实现创建新用户。

```
[root@365linux ~]# su - demo
[demo@365linux ~]$ useradd user05
-bash: /usr/sbin/useradd: 权限不够
[demo@365linux ~]$ sudo useradd user05
[sudo] password for demo:
```

使用 `sudo` 时需要确认用户的身份，此时输入的是 `demo` 用户自身的密码，而不是 root 用户的密码。这样既可以给普通用户授予管理系统的权限，又避免了泄露 root 用户的密码。而且在必要时，容易根据不同的用户记录追踪用户的操作记录。

## 6.4 用户账号初始化

### 1. 用户特定配置文件

当用户被创建时，系统从 `/etc/skel/` 目录下复制用户的配置文件到用户的家目录。这些配置文件用来定义用户的工作环境，比如 `PATH` 路径、命令别名等。这些文件位于每个用户的家目录内，所以只对当前用户有效。





- `~/.bashrc` : 定义函数和别名。
- `~/.bash_profile` : 设置环境变量。
- `~/.bash_logout` : 定义用户退出时执行的命令。



### REAL-WORK TIP

对于要在系统中经常执行的复杂的命令，可以使用别名的方式将其输入简化。为了保证下一次启动依然有效，可将别名定义在 `bashrc` 文件中。

## 2. 全局用户配置文件

顾名思义，这些是对所有用户都生效的设置。

- `/etc/bashrc` : 定义函数和别名。
- `/etc/profile` : 设置环境变量。
- `/etc/profile.d` : 目录下的脚本被 `/etc/profile` 引用。

## 3. 系统预设的值

用户属性信息的默认值在文件 `/etc/login.defs` 中被定义。

**示例：**查看 `/etc/login.defs`，使用 `grep` 命令过滤掉文件中以“#”开头的注释行。

```
[root@365linux ~]# grep -v '^#' /etc/login.defs
MAIL_DIR    /var/spool/mail
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_MIN_LEN     5
PASS_WARN_AGE    7

UID_MIN                500
UID_MAX                60000
GID_MIN                500
GID_MAX                60000
CREATE_HOME            yes
UMASK                  077
USERGROUPS_ENAB        yes
ENCRYPT_METHOD          SHA512
```

创建的新用户的默认属性信息在配置文件 `/etc/default/useradd` 中被定义。

**示例：**查看 `/etc/default/useradd` 文件中的配置项。

```
[root@365linux ~]# cat /etc/default/useradd
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
```



```
SKEL=/etc/skel  
CREATE_MAIL_SPOOL=yes
```

**示例：**使用 `useradd` 命令列出系统预设的添加用户信息的默认值。

```
[root@365linux ~]# useradd -D  
GROUP=100  
HOME=/home  
INACTIVE=-1  
EXPIRE=  
SHELL=/bin/bash  
SKEL=/etc/skel  
CREATE_MAIL_SPOOL=yes
```

命令输出结果分别表示新建用户的组群 ID 初始值；用户家目录的上一级目录；账号过期后失效时间；账号过期日期；默认的登录 Shell；用户默认配置文件的源目录（从指定的目录中复制配置文件到用户的家目录）；默认创建的用户邮箱文件。

可以通过 `useradd -D` 加上相应的选项更新这些预设值。

`useradd -D` 可使用的选项如下。

- `-b`：定义用户家目录的上一级目录。
- `-e`：用户账号的过期日期。
- `-f`：用户账号过期几日后失效。
- `-g`：新建用户的起始群组或 ID。
- `-s`：新建用户登录后使用的 Shell。



## 本章总结

本章主要讲解了 Linux 用户和组的管理。用户和组是 Linux 系统的基本概念，是实现多用户协作、权限分配、文件共享等任务的基础。理解普通用户和 root 用户的特点，在需要时切换到 root 用户或者获取管理权限。

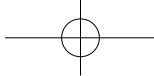
在实际工作中，管理员要对系统全局环境或者用户的特定文件进行设置，以满足项目对于用户的需求。比如设置所有新建用户自动属于某个特定的组。

当需要一次性重复创建很多账号时，可以编辑 Shell 脚本批量处理。

不管实际应用如何变化，用户和组的基本原则是不变的。

### 章节重点：

- 了解用户的基本概念，RHEL 系统用户类型。
- 使用图形界面用户管理工具查看、添加、修改、删除用户。
- 使用命令行工具查看、添加、修改、删除用户。



- 理解添加用户时，系统的实现过程。
- 了解组的基本概念，添加和删除组。



## 基础技能演练

### 1. 回答以下问题：

- (1) 什么命令（选项）能够用来创建一个用户 JSmith，并且描述为“Jr Admin”？
- (2) /etc/shadow 每个字段的意义是什么？
- (3) 哪个命令可以创建组 sales？怎样把用户 JSmith 添加到这个组？
- (4) 有什么方法可以提升普通用户的权限来运行系统命令？
- (5) 如果你想在用户 user01 登录系统的时候自动执行一个命令，编辑哪个文件？
- (6) 当一个新用户被创建时，默认会在用户的家目录下创建哪些文件？
- (7) 默认情况下，用户的家目录位于系统的哪个目录？

### 2. 用户和组的管理。

- (1) 创建用户和组，要求如下。

用户名：user01；密码：password；家目录：/home/user01；Shell：BASH；描述：“Normal User”；UID：600；GID：800；user01 是 sales 组的成员。

- (2) 创建一个新的用户，要求如下。

用户名：user02；密码：secret；家目录：/var/user02；Shell：BASH；描述：“Auditor”；UID：601；GID：800；user02 是 sales 组的成员。

- (3) 创建一个新的用户，要求如下。

用户名：user03；密码：supercool；家目录：none；Shell：none；描述：“Fake Account”；UID：602；GID：801；user03 是 engineering 组的成员。

- (4) 创建一个新的用户，要求如下。

用户名：user04；密码：CheaperThanFree；家目录：/home/user04；Shell：csh；描述：“Developer”；UID：603；GID：802；user04 是 developers 组的成员。

- (5) 创建一个新的用户，要求如下。

用户名：user05；密码：S3cuReP；家目录：none；Shell：none；描述：“Security User”；user05 是一个系统用户。



## 提高技能实践

### 1. 自定义 Profile。

- (1) 创建一个文件 hr\_notice.txt，内容如下：



```
Human Resource Notice
All employees are subject to random drug tests.
Sincerely,
HR
```

(2) 设置当系统创建新的用户时，`hr_notice.txt` 文件会被复制到它们的家目录。

2. 完成如下操作：

(1) 修改用户 `user01` 的用户名为 `user001`，登录的 Shell 为 `csh`。

(2) 修改 `user001` 的密码有效期为 30 天，提前 3 天警告，密码过期后 2 天账号失效，并设置账号永久失效日期为下月的 28 日。

(3) 将用户 `user02` 加入到组 `developers`，并使其成为组的管理员。

(4) 切换到用户 `user03`，使其临时切换主组为 `developers`。

(5) 使用 3 种以上的方法限制 `user004` 登录，测试比较它们的区别。

3. 根据添加用户时系统实现过程的解析，不使用 `useradd`、`passwd`、`groupadd` 命令，仅使用 `vim`、`cp`、`touch`、`grub-crypt`（生成加密码）、`chown`、`chmod` 等命令创建一个用户，测试其是否能够正常登录。

# Linux文件权限管理

通过文件的权限控制用户对文件的访问。Linux 文件权限系统简单而又灵活，易于理解和应用，可以轻松地处理最常见的权限情况。文件只具有 3 种应用权限的用户类别和 3 种基本控制权限。

应用权限的用户类别：

- 文件拥有者。文件归用户所有，通常是创建文件的用户，但可以更改。
- 文件所属组。文件归单个组所有，通常为创建该文件的用户的主组，但可以更改。
- 其他用户。除了拥有者，所属组外的其他用户。

应用权限时，用户权限优先级高于组的权限，高于其他人的权限。

## NOTE

每个用户都有自己所属的组，每个文件也有所属的组，这两个组的意义是不同的。文件所属组可以恰好是该文件拥有者的所属组，也可以是另一个不同的组。

3 种基本控制权限：读取、写入和执行。这些权限的作用如表 7-1 所示。

表7-1 文件和目录的权限

权 限	对文件的影响	对目录的影响
r (读取)	可以读取文件的内容	可以列出目录的内容（文件名）
w (写入)	可以更改文件的内容	可以创建或删除目录中的任一文件
x (执行)	可以作为命令执行文件	可以访问目录的内容（还取决于目录中文件的权限）

## NOTE

访问并进入目录需要用户同时对目录有读取和执行的权限。如果用户仅对某目录有读取的访问权限，则可以列出其中文件的名称，但是其他信息（包括权限或时间戳）都不可用，也不可访问。如果用户仅对某目录具有执行的权限，则用户不能列出该目录中文件的名称。但是如果用户已经知道对其具有读取权限的文件的名称，那么可以通过明确指定文件名来访问该目录下指定文件的内容。



默认情况下，如果用户对某个目录具有写入的权限，那么他可以删除该目录下的任何文件，不论被删除文件的拥有者是谁、权限设置如何。

**示例：**在命令行中列出文件的属性。

```
[demo@365linux ~]$ ll /etc/man.config
-rw-r--r--. 1 root root 4940 11月 17 2012 /etc/man.config
文件类型与权限 链接数 拥有者 所属组 文件大小 文件被创建或修改日期 文件名
```

除去文件类型的标识位，文件权限字段一共有 9 个符号，每 3 个符号为一组，分别表示文件拥有者的权限、所属组的权限和其他人的权限。如示例中的文件，拥有者为 root 用户，他的权限为 rw-（可读、可写、执行权限位上没有执行权限）；所属组为 root 组，所属组的权限为 r--（仅有可读的权限）；其他人也是仅有可读的权限。

**示例：**在命令行中列出目录的属性。

```
[demo@365linux ~]$ ll -d /home/demo/
drwx-----. 29 demo demo 4096 3月 24 04:48 /home/demo/
```

该目录的权限为，只有拥有者 demo 用户对它有读、写、执行的权限，所属组和其他人都没有任何权限。这是用户家目录的特点，即只有用户本身才能访问自己的家目录。

## 7.1 使用GUI工具管理权限

Nautilus 文件管理器允许用户使用最常见的配置对文件的基本权限进行设置或更改。

打开 Nautilus 文件管理器，在选定的文件或文件夹上单击鼠标右键，在弹出的快捷菜单中选择“属性”命令；在弹出的对话框中单击“权限”选项卡，即文件权限的配置界面，如图 7-1 所示。



图7-1 文件属性“权限”



在 Nautilus 文件浏览器中，文件或目录的图标外观区别反映出当前用户对文件是否有访问的权限，如图 7-2 所示。

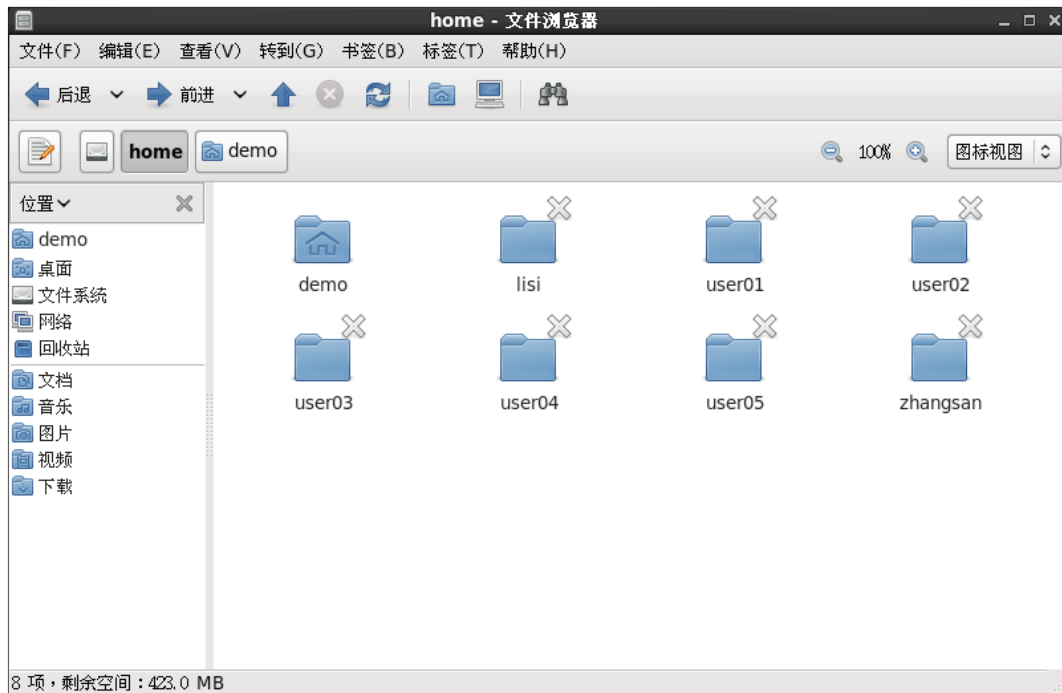


图7-2 Nautilus文件浏览器中的文件图标

## 7.2 在命令行中管理权限

前文已经示例在命令行中查看文件或目录的权限。

要更改文件的权限，可使用 `chmod` 命令。

### 1. 使用 `chmod` 命令更改文件权限

用法：`chmod [选项]... 模式 [, 模式]... 文件 ...`

`chmod [选项]... 八进制模式 文件 ...`

`chmod [选项]... --reference= 参考文件 文件 ...`

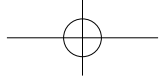
常用选项：

`-R`：以递归方式更改所有的文件及子目录。

符号模式：`chmod WhoWhatWhich file|directory`

简单理解：`chmod [-R] [ugoa][+=[rwx] 文件 | 目录`

- Who 是指 u、g、o、a（代表拥有者、组、其他、全部）。



- What 是指 +、-、= (代表添加、删除、精确设置)。
- Which 是指 r、w、x (代表读取、写入、可执行的权限)。

八进制模式：`chmod [-R] ### file|directory`

- ### 是指三位数字，每一个数字代表一个权限对象类别：拥有者、组、其他人。
- # 一位数字是指在该对象（比如拥有者）上的权限的总和， $r=4$ 、 $w=2$ 、 $x=1$ 。那么如果文件的拥有者的权限是 `rwX`，用数字表示就是 7（即  $4+2+1=7$ ）。

示例：设置 `file` 文件的拥有者加上执行的权限，其他人去除读取的权限。

```
[demo@365linux ~]$ ll file
-rw-rw-r--. 1 demo demo 267 3月 19 12:45 file
[demo@365linux ~]$ chmod u+x,o-r file
[demo@365linux ~]$ ll file
-rwxrw----. 1 demo demo 267 3月 19 12:45 file
```

注意文件前后的权限变化，当文件有了可执行权限时，文件在终端中的颜色会高亮。

示例：精确设置文件的权限。

```
[demo@365linux ~]$ chmod u=rw,g=r file
```

设置 `file` 文件的拥有者的权限为可读、可写，组仅可读，其他人没有权限。

示例：统一设置目录及目录下所有文件的权限。

```
[demo@365linux ~]$ chmod -R +w targetdir/
```

给目录 `targetdir` 及该目录下的所有文件在所有用户类型权限位上添加可写的权限。

示例：使用数字方式设置文件的权限。

```
[demo@365linux ~]$ chmod 755 file
```

设置文件 `file` 的权限为 `rwXr-Xr-X`，即拥有者可读、可写、可执行权限，组和其他人都为可读、可执行权限。

## 2. 使用 `chown` 命令更改文件用户所有权

要更改文件或文件夹的用户或组的所有权，使用 `chown` 或 `chgrp` 命令。

用法：`chown [选项]... [所有者][:[组]] 文件...`

`chown [选项]... --reference= 参考文件 文件...`

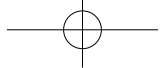
示例：改变文件 `file` 的所有者。

```
[root@365linux demo]# ll file
-rwxr-xr-x. 1 demo demo 267 3月 19 12:45 file
[root@365linux demo]# chown zhangsan file
[root@365linux demo]# ll file
-rwxr-xr-x. 1 zhangsan demo 267 3月 19 12:45 file
```

示例：改变文件 `file` 的所属组。

```
[root@365linux demo]# chgrp mail file
[root@365linux demo]# ll file
-rwxr-xr-x. 1 zhangsan mail 267 3月 19 12:45 file
```





**示例：**同时改变用户和组的所有权，并使用 **-R** 选项递归目录和目录下的所有文件。

```
[root@365linux demo]# chown zhangsan:mail targetdir/
```



#### NOTE

在变更文件或目录的用户和组的所有权时，前提条件是指定的目标用户和组在系统中已经存在。

## 7.3 特殊权限

系统中有些地方的设计需要一些特殊的权限。比如 **passwd** 命令，普通用户在使用 **passwd** 命令修改自己的密码时，也需要更新 **/etc/shadow** 文件，而该文件只有管理员可以修改；这就意味着，普通用户成功修改了自己的密码，那么他在运行 **passwd** 命令时获得了管理员的权限或者说是使用 **root** 的身份在执行。这是怎么做到的呢？

**示例：**查看 **passwd** 命令的权限。

```
[demo@365linux ~]$ ll `which passwd`  
-rwsr-xr-x. 1 root root 25980 2月 17 2012 /usr/bin/passwd
```

可以看到在 **root** 用户的权限位上是 **rws**，这表示在执行权限位上，除了 **x** 还有一个 **setuid** 的权限，这个就是特殊权限的一种。

特殊权限对文件和目录的影响如表 7-2 所示。

表7-2 特殊权限对文件和目录的影响

特殊权限	对文件的影响	对目录的影响
u+s (setuid 或者 suid)	以拥有文件的用户身份执行文件，而不是以运行文件的用户身份	无影响
g+s (setgid 或者 sgid)	以拥有文件的组身份执行文件	在目录中最新创建的文件将其组所有者设置为与目录的组所有者相同
o+t (sticky)	无影响	对目录具有写入权限的用户仅可以删除其拥有的文件，而无法删除其他用户所拥有的文件

**示例：**含有 **sgid** 权限的文件。

```
[root@365linux ~]# ll /usr/bin/wall  
-r-xr-sr-x. 1 root tty 10932 6月 18 2013 /usr/bin/wall
```

**示例：**含有 **sticky** 权限的目录。

```
[root@365linux ~]# ll -d /tmp  
drwxrwxrwt. 30 root root 4096 3月 26 08:17 /tmp
```

**NOTE**

因为特殊权限和执行权限在同一个位置上，所以特殊权限的大小写表示了该位置上是否还有可执行权限。如果特殊权限为小写，则表示此处有 x 权限；如果特殊权限为大写，则表示此处没有 x 权限。不过从特殊权限的应用角度来看，其总是要伴随 x 权限一起使用才有实际意义。

设置特殊权限有两种方法。

- 用符号法设置：setuid = u+s；setgid = g+s；sticky = o+t。
- 用数字法设置：setuid = 4；setgid = 2；sticky = 1。

示例：给目录设置 sgid 权限。

```
[root@365linux ~]# chmod u+s directory
或者：
[root@365linux ~]# chmod 2775 directory
```

**REAL-WORK TIP**

因为 setuid 可以让普通用户拥有 root 的身份和权限，所以为了安全起见，您应该留意系统中含有 setuid 位的程序或脚本是否异常。可以通过 find 命令从系统中找出它们：

```
[root@365linux ~]# find / -perm -4000
```

## 7.4 隐藏的扩展属性（权限）

为了极大地保证文件的安全，Linux 文件系统为文件预留了一些扩展的属性，比如设置让 root 用户也无法删除文件的权限。这些属性被隐藏了起来，需要特定的命令才能查看和设置。

示例：查看文件的扩展属性。

```
[root@365linux ~]# lsattr testfile.txt
-----e- testfile.txt
```

“e” 就是一个扩展属性，表示使用磁盘块映射。这是一个默认的属性，不能去除。

示例：设置文件的扩展属性。

```
[root@365linux ~]# chattr +i testfile.txt
[root@365linux ~]# lsattr testfile.txt
----i-----e- testfile.txt
```

给文件添加 “i” 的属性表示该文件不能被删除、不能修改、不能重命名、不能创建链接。更多的文件扩展属性设置选项可以通过 man chattr 来获得详细的说明。



## 7.5 访问控制列表（ACL）

Linux 系统通常的权限管理只是针对文件或目录的拥有者、所属组，其他人只进行读、写、执行的权限的划分。如果要对某个文件或目录进行除了上述三类用户的单一特定用户或组的更细致的权限划分，比如，针对某个文件，除拥有者和组之外，其他人都没有读取的权限，而唯独用户 zhangsan 需要读取的权限（zhangsan 属于其他人的范围），这就需要借助 ACL（Access Control List）进行。

ACL 的使用需要文件系统的支持，目前绝大部分的 Linux 文件系统（如 EXT2/3/4、JFS、XFS）均支持 ACL 的功能。在 RHEL 中 ACL 是默认启动的。如果系统默认没有启动 ACL 的功能，则需要添加 ACL 属性并重新挂载文件系统以获取 ACL 的支持（后面的章节介绍文件系统挂载的操作）。

在确定某个进程是否能够访问某一文件时，权限的优先级如下：

- （1）如果是文件的拥有者身份运行该进程，那么就应用该文件的拥有者权限；
- （2）如果是列于用户 ACL 条目中的用户运行该进程，那么就应用用户 ACL（只要 mask 允许）。
- （3）如果是文件的所属组身份或具有明确组 ACL 条目的组身份运行该进程，且权限是由任意匹配组授予的，则应用组的权限（只要 mask 许可）。
- （4）否则，应用文件的其他权限。

其中 mask 称作具有 ACL 的文件的掩码，用于限制组成员、ACL 补充用户和组成员的最大权限。

ACL 的常用命令介绍如下。

### 1. getfacl 命令

描述：查看文件或目录的 ACL 权限。

用法：getfacl [options] file ...

示例：查看文件的 ACL。

```
[root@365linux test]# getfacl testfile
# file: testfile
# owner: root
# group: root
user::rw-
user:zhangsan:rwx
group::r--
mask::rwx
other::r--
```

文件 testfile 的用户和组都是 root，它的权限是 644，只有 root 用户拥有读、写权限，其他人都是只拥有只读权限。而设置了 ACL 后，ACL 用户 zhangsan 则对该文件拥有读、写、执行的权限。



对于添加了 ACL 的文件，使用 `ls -l` 命令列出时，权限位的最后带有 “+” 号，如下：

```
[root@365linux test]# ll testfile
-rw-rwxr--+ 1 root root 75 3月 26 09:33 testfile
```

## 2. setfacl 命令

描述：设置某个文件或目录的 ACL 权限。

用法：`setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...`

常用选项：

- `-m`：设置或修改文件的 ACL 权限。
- `-x`：取消文件的一个 ACL 权限。
- `-b`：删除文件所有的 ACL 权限。
- `-k`：删除所有的默认的 ACL 权限。
- `--set`：设置文件的 ACL，替代当前的 ACL。
- `--mask`：重新计算有效的 mask 值。
- `-R`：递归子目录。
- `-d`：设置默认的 ACL 权限，仅能针对目录使用。
- `--restore`：从文件恢复备份的 ACL。

### 1) 查看和设置 ACL

示例：使用 `setfacl` 命令设置文件的 ACL。

**Step 01** 针对用户 `zhangsan` 来设置权限为 `rw`。

```
[root@365linux test]# setfacl -m u:zhangsan:rw acltest.file
```

**Step 02** 针对组 `sales` 来设置权限为 `rw`。

```
[root@365linux test]# setfacl -m g:sales:rw acltest.file
```

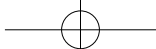
**Step 03** 设置限制的权限为 `r`。

```
[root@365linux test]# setfacl -m m:r acltest.file
```

**Step 04** 查看文件当前的 ACL。

```
[root@365linux test]# getfacl acltest.file
# file: acltest.file
# owner: root
# group: root
user::rw-
user:zhangsan:rw-      #effective:r--
group::r--
group:sales:rw-        #effective:r--
mask::r--
other::r--
```

最终的权限：文件的拥有者 `root` 的权限为 `rw-`；ACL 用户 `zhangsan` 的权限为 `rw`，但因



为设置了限制权限 `mask` 为 `r--`，两个权限相与后，最终 `zhangsan` 对该文件的权限为 `r--`；同理，文件所属组的权限为 `r--`；ACL 组 `zhangsan` 的权限受 `mask` 的影响也为 `r--`；其他人的权限为 `r--`。

设置 ACL 权限，如果同时设置多个权限，权限之间使用 “,” 分隔。

**示例：**同时设置多个 ACL 用户和组的权限。

```
[root@365linux test]# setfacl -m u:user03:rwx,u:user04:rwx,g:sales:rw testfile
```

## 2) 删除文件的 ACL 权限

**示例：**删除 ACL 权限。

```
[root@365linux test]# setfacl -x g:zhangsan acltest.file
```

示例中删除了 ACL 组的权限，删除 ACL 用户权限的操作类似。删除后可执行 `getfacl` 命令查看结果。

**示例：**一次性删除所有的 ACL 权限。

```
[root@365linux test]# setfacl -b acltest.file
```

## 3) 设置目录默认的 ACL

如果希望在一个目录中新建的文件和子目录都使用同一个预定的 ACL，那么我们可以使用默认（Default）ACL。在对一个目录设置了默认的 ACL 以后，每个在目录中创建的文件都会自动继承目录的默认 ACL 作为自己的 ACL。

**示例：**设置目录默认的 ACL。

**Step 01** 设置目录 `testdir` 的 ACL 指定组 `sales` 的权限为 `rwx`。

```
[root@365linux test]# setfacl -d -m g:sales:rwx testdir/
```

**Step 02** 查看目录 `testdir` 当前的 ACL 权限。

```
[root@365linux test]# getfacl testdir/
# file: testdir/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:sales:rwx
default:mask::rwx
default:other::r-x
```

**Step 03** 在目录 `testdir` 中创建一个文件 `file.txt`。

```
[root@365linux test]# touch testdir/file.txt
```

**Step 04** 查看该文件的权限，自动继承了上级目录的 ACL（受文件默认的 `mask` 影响）。

```
[root@365linux test]# getfacl testdir/file.txt
```



```
# file: testdir/file.txt
# owner: root
# group: root
user::rw-
group::r-x                #effective:r--
group:sales:rw            #effective:rw-
mask::rw-
other::r--
```

#### 4) 备份和恢复 ACL

主要的文件操作命令 **cp** 和 **mv** 都支持备份时保留文件的 ACL, **cp** 命令需要加上 **-p** 参数。但是 **tar** 等常见的备份工具是不会保留目录和文件的 ACL 信息的。这种情况下, 如果备份和恢复带有 ACL 的文件及目录, 那么可以先把文件的 ACL 权限信息备份到一个文件里, 以后用 **--restore** 选项来恢复这个文件中保存的 ACL 信息。

**示例:** 文件的 ACL 信息备份和恢复。

**Step 01** 查看目录及其所有子目录和文件当前的 ACL 信息。

```
[root@365linux test]# getfacl -R testdir/
# file: testdir/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:sales:rwx
default:mask::rwx
default:other::r-x

# file: testdir//file.txt
# owner: root
# group: root
user::rw-
group::r-x                #effective:r--
group:sales:rw            #effective:rw-
mask::rw-
other::r--
```

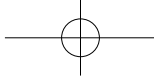
**Step 02** 备份目录及其子目录中文件的 ACL。

```
[root@365linux test]# getfacl -R testdir/ > testdir.acl
```

**Step 03** 为测试, 删除原文件所有的 ACL。

```
[root@365linux test]# setfacl -R -b testdir/
```

**Step 04** 查看删除后的权限。



```
[root@365linux test]# getfacl -R testdir/
# file: testdir/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x

# file: testdir//file.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

**Step 05** 从 testdir.acl 文件中恢复被删除的 ACL 信息。

```
[root@365linux test]# setfacl --restore testdir.acl
```

**Step 06** 查看恢复后的效果。

```
[root@365linux test]# getfacl -R testdir
.....和备份前一样。
```

## 7.6 SELinux前瞻

SELinux（安全增强性 Linux）提供了一种在传统的系统安全设置之外的额外机制，它的设置会影响用户、进程对系统资源的访问。

在某种程度上，它可以被看作与标准权限并行的权限系统。在常规权限模式中，以用户身份运行进程，并且系统上的文件和其他资源都设置了权限（控制哪些用户对哪些文件具有哪些访问权限）标签。SELinux 通过可配置的策略控制哪些进程可以访问哪些文件。若要访问文件，用户或进程必须同时具有普通访问权限和 SELinux 访问权限。因此，即使以管理员用户身份 root 运行进程，根据进程及文件或资源的 SELinux 安全上下文也可能被拒绝访问目标文件或资源。

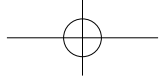
SELinux 以如此严格的安全限制策略来防范应用程序未知漏洞可能对系统造成的侵害。

对于 SELinux 的权限配置涉及的操作超出了本章节的难度，在此仅做简要介绍，将在后续的章节中涉及。

RHEL 系统默认情况下开启了 SELinux。在实验环境中，如果碰到 SELinux 权限受阻而又无法正确配置时，可将其禁用。禁用的方式有两种：临时禁用和永久禁用。

**示例：**临时禁用 SELinux 安全机制，手动恢复或重启后重新生效。

**Step 01** 查看当前 SELinux 开关。



```
[root@365linux ~]# getenforce
Enforcing
```

Enforcing 表示当前 SELinux 开启。

### Step 02 临时失效 SELinux。

```
[root@365linux ~]# setenforce 0
[root@365linux ~]# getenforce
Permissive
```

使用 setenforce 命令设置为 0，即 Permissive（宽容）模式，SELinux 安全机制不生效（仍会提示权限问题）。

### Step 03 恢复 SELinux。

```
[root@365linux ~]# setenforce 1
[root@365linux ~]# getenforce
Enforcing
```

示例：永久禁用 SELinux。

### Step 01 修改 SELinux 的配置文件。

```
[root@365linux ~]# vim /etc/selinux/config
找到 SELINUX=enforcing
修改为 SELINUX=disabled
```

### Step 02 重启系统使配置生效。

```
[root@365linux ~]# reboot
```



## REAL-WORK TIP

在生产环境中，特别是在面对外部网络的服务器时，建议保持 SELinux 开启，以提高 Linux 系统的安全级别。对 SELinux 配置文件 /etc/selinux/config 的修改要非常小心，一旦配置错误，重启后会因直接导致内核错误而无法启动系统。



## 本章总结

因为 Linux 系统的基本权限设置了用户、组、其他 3 种所有者类型，读、写、执行 3 种简单明了的文件权限管理模式，所以对 Linux 文件权限管理是比较简洁而且容易配置的。实际应用中，对于文件权限的理解非常重要，它是 Linux 系统对于系统资源和进程之间关系的基础。在后续对于系统应用、服务的配置过程中，程序无法启动、服务无法正常提供资源，很多情况下可能是由文件的权限配置不正确引起的，因此需要优先排查。而粗放的文件权限管理则会给系统的安全埋下隐患。

本章需要重点理解和掌握 Linux 基本权限的管理；对于特殊权限、扩展属性、ACL 适当了解，因为它们在通常情况下，默认不会被开启使用。





### 章节重点：

- 理解 Linux 系统文件权限设计。
- 使用命令行工具管理权限。

## 基础技能演练

为多用户创建共享工作目录。

在系统中创建 3 个账号：liuwei、licheng、zhangming，这些账号都是组 technology 的成员。

创建名为 /home/resource 的目录。修改此目录的权限，使目录的拥有者为 zhangming，所属组为 technology，用户和组成员都可以访问、创建和删除该目录中的文件；除此之外，其他人有只读的权限。在此目录中创建的文件应该自动被分配到属于 stooges 组。

## 提高技能实践

使用 ACL 授予和限制访问权限。

此实验开始之前需要在服务器上创建需要的用户和组。

研究生需要名为 /opt/research 的协作目录，用于存储他们的研究成果。只有组 profs 和 grads 中的成员能够在该目录中创建新文件，并且新文件应具有以下属性：

- 目录应归用户 root 所有。
- 新文件应归组 grads 所有。
- 教授（组 profs 的成员）应自动拥有对新文件的读、写访问权限。
- 暑期实习生（组 interns 的成员）应自动拥有对新文件的只读访问权限。
- 其他用户（不是组 profs、grads 或 interns 的成员）绝对不能拥有对该目录及其内容的访问权限。



## 第8章

# 建立网络连接

一台 Linux 系统的主机，不管是作为桌面系统、工作站还是服务器，大多数情况下，它都需要连接到网络。作为客户端，还是服务器端；连接到内网（局域网），还是外网（广域网）；使用静态 IP，还是动态获取地址；如果您明确知道这些名词的概念，那么是配置 Linux 系统网络的方法是非常容易的，甚至在拥有 DHCP（动态主机设置协议）服务的网络环境中，Linux 系统已经完成自动配置并连接到网络。然而我们还是要深刻理解这些操作背后的原理和逻辑。

先来了解一下基本网络概念（以下概念相关内容部分摘自维基百科）。

## 8.1 基本网络概念

### 1. 什么是网络

网络是指利用通信设备和线路将地理位置不同、功能独立的多个主机系统连接起来，以功能完善的网络软件实现网络上的硬件、软件及资源的共享和信息的传递。简单来说，网络即连接两台或多台主机进行通信的系统。这里的“主机”通常是指计算机，但也不一定，它也可以是打印机、智能手机等。随着技术的发展，将来会有越来越多不同类型的设备可连接到网络。

### 2. 网络类型

网络按照其覆盖的范围可以分成局域网和广域网。

局域网（Local Area Network，LAN）又称内网，是指覆盖局部区域（如办公室或楼层）的计算机网络。

广域网（Wide Area Network，WAN）又称外网、公网，是指连接不同地区的局域网或城域网的计算机通信的远程网。通常跨接很大的物理范围，所覆盖的范围从几十千米到几千千



米，它能连接多个地区、城市和国家，或横跨多个洲，并能提供远距离通信，形成国际性的远程网络。广域网概念上并不等于互联网。

网络按照交换方式，又可分为以太网、公用电话交换网、分组交换网、令牌环网等。



#### NOTE

网络是一个宽泛的概念，并不一定是指计算机网络。对于 Linux 系统的学习，从应用角度出发，我们通常所说的局域网一般情况下指内部计算机以太网（可能涉及光纤网络），而外网通常是指连接到外部的互联网。

### 3. 互联网相关协议

有关互联网的协议可以分为 3 层。

最底层的是 IP（Internet Protocol，即网际协议），是用于报文交换网络的一种面向数据的协议，这一协议定义了数据包在网际传送时的格式。目前使用最多的是 IPv4 版本，这一版本中用 32 位定义 IP 地址，尽管地址总数达到 43 亿，但是仍然不能满足现今全球网络飞速发展的需求，因此 IPv6 版本应运而生。在 IPv6 版本中，IP 地址共有 128 位，“几乎可以为地球上每一粒沙子分配一个 IPv6 地址”。IPv6 目前并没有普及，许多互联网服务提供商并不支持 IPv6 协议的连接。但是，可以预见，将来在 IPv6 的帮助下，任何智能设备都有可能连入互联网。

上一层是 UDP 协议和 TCP 协议，它们用于控制数据流的传输。UDP 是一种不可靠的数据流传输协议，仅为网络层和应用层之间提供简单的接口。而 TCP 协议则具有高可靠性，通过为数据报加入额外信息，并提供重发机制，它能够保证数据不丢包、没有冗余包及保证数据报的顺序。对于一些需要高可靠性的应用，可以选择 TCP 协议；相反，对于性能优先考虑的应用如流媒体等，则可以选择 UDP 协议。

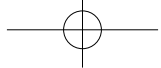
最顶层的是一些应用层协议，这些协议定义了一些用于通用应用的数据报结构，包括以下几种。

- DNS：域名服务。
- FTP：服务使用的是文件传输协议。
- HTTP：所有的 Web 页面服务均使用超级文本传输协议。
- POP3：邮局协议。
- SMTP：简单邮件传输协议。
- Telnet：远程登录等。

### 4. OSI 参考模型与 TCP/IP 参考模型

#### 1) OSI 参考模型

开放系统互连参考模型（Open System Interconnection Reference Model, ISO/IEC 7498-1），



简称为 OSI 模型（OSI Model），是一种概念模型，由国际标准化组织（ISO）提出，是一个试图使各种计算机在世界范围内互连为网络的标准框架。

OSI 将计算机网络体系结构划分为 7 层，如图 8-1 所示。

### 第 7 层 应用层

应用层能与应用程序界面沟通，以达到展示给用户的目的。在此常见的协议有 HTTP、HTTPS、FTP、Telnet、SSH、SMTP、POP3 等。

### 第 6 层 表示层

表示层能为不同的客户端提供数据和信息的语法转换内码，使系统能解读成正确的数据，同时也能提供压缩解压、加密解密。

### 第 5 层 会话层

会话层用于为通信双方制定通信方式，并创建、注销会话（双方通信）。

### 第 4 层 传输层

传输层用于控制数据流量，并且进行调试及错误处理，以确保通信顺利。而传送端的传输层会为分组加上序号，方便接收端把分组重组为有用的数据或文件。

### 第 3 层 网络层

网络层的作用是决定如何将发送方的数据传到接收方。该层通过考虑网络拥塞程度、服务质量、发送优先权、每次路由的耗费来决定节点  $X$  到节点  $Y$  的最佳路径。我们熟知的路由器就工作在这一层，通过不断地接收与传送数据使得网络变得相互连通。

### 第 2 层 数据链路层

首先，数据链路层的功能在于管理第一层的比特数据，并且将正确的数据传送到没有传输错误的路线中。其作用是处理由数据受损、丢失甚至重复传输错误造成的问题，使后续层级不受影响。设备有 Bridge 和 switch。

### 第 1 层 物理层

物理层定义了所有电子及物理设备的规范。其中特别定义了设备与物理媒介之间的关系，这包括了针脚、电压、线缆规范、集线器、中继器、网卡、主机适配器（在 SAN 中使用的主机适配器）及其他设备的设计定义。因为物理层传送的是原始的比特数据流，即设计的目的是为了保证当发送时的信号为二进制数“1”时，对方接收到的也是二进制数“1”而不是二进制数“0”。因而就需要定义哪个设备有几个针脚，其中哪个针脚发送的多少电压代表二进制数“1”或二进制数“0”；以及一个 bit 需要持续几微秒，传输信号是否在双向上同时进行，最初的连接如何创建和最终如何终止等问题。

## 2) TCP/IP 参考模型

TCP/IP 参考模型是一个抽象的分层模型，这个模型中，所有的 TCP/IP 系列网络协议都被归类到 4 个抽象的“层”中，如图 8-1 所示。每一抽象层建立在低一层提供的服务上，并



且为高一层提供服务。

#### 第4层 应用层

该层包括所有和应用程序协同工作，利用基础网络交换应用程序专用的数据的协议。常用的应用层协议有 HTTP、FTP、POP3、SMTP、Telnet、SSH、BOOTP、NTP、DNS、ECHO、SNMP、DHCP、ARP。

#### 第3层 传输层

传输层的协议能够解决诸如端到端可靠性（数据是否已经到达目的地）和保证数据按照正确的顺序到达这样的问题。在 TCP/IP 协议组中，传输协议也包括所给数据应该传送给哪个应用程序。相关的协议如 TCP、UDP、DCCP、RTP。

#### 第2层 网络互连层

网络互连层解决在一个单一网络上传输数据包的问题。

对于 TCP/IP 来说这是网际协议(IP)(如 ICMP 和 IGMP 这样的必需协议尽管运行在 IP 上，也仍然可以看作网络互连层的一部分；ARP 不运行在 IP 上)。

#### 第1层 网络接口层

网络接口层实际上并不是因特网协议组中的一部分，但是它是数据包从一个设备的网络层传输到另外一个设备的网络层的方法。

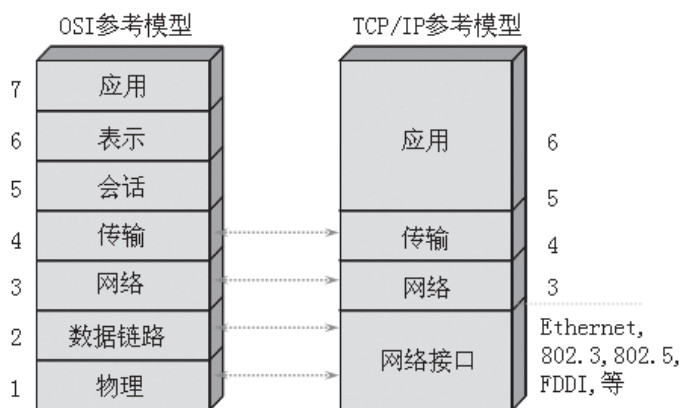
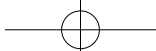


图8-1 TCP/IP参考模型与OSI参考模型映射关系

### 5. IPv4 地址

IPv4 地址是一个 32 位数字、通常使用点号分隔的 4 个十进制数，具有八位字节（取值范围为 0 ~ 255）表示。该地址分为两个部分：网络位和主机位。网络位用来标识子网，具有相同网络位的 IP 地址称为属于同一子网，同一子网中的所有主机可以在彼此之间直接通信，无须路由器。主机位用来标识子网中的特定主机，同一子网中每台主机的主机位必须是唯一的。

子网的大小是可变的。网络管理员通过指定子网掩码来指明多少位的 IPv4 地址属于同一子网。可供主机位使用的位数越多，子网中可用的主机数就越大。



子网掩码可用两种格式表示，比如 255.255.255.0 和 /24 表示同样的值。

将子网中最低的地址（主机位二进制值全部为 0）称为网络地址；子网中最高的地址（主机位二进制值全部为 1）用于广播消息，称为广播地址。

IP 地址、子网掩码、网络地址及广播地址的关系如表 8-1 至表 8-3 所示。

表8-1 192.168.1.107/24

主机地址	192.168.1.107	11000000.10101000.00000001.01101011
子网掩码	255.255.255.0 或 /24	11111111.11111111.11111111.00000000
网络地址	192.168.1.0	11000000.10101000.00000001.00000000
广播地址	192.168.1.255	11000000.10101000.00000001.11111111

表8-2 172.16.0.123/16

主机地址	172.16.0.123/16	10101100.00010000.01111011
子网掩码	255.255.0.0 或 /16	11111111.11111111.00000000.00000000
网络地址	172.16.0.0	10101100.00010000.00000000.00000000
广播地址	172.16.255.255	11000000.10101000.11111111.11111111

表8-3 10.1.1.18/8

主机地址	10.1.1.18/8	00001010.00000001.00000001.00010010
子网掩码	255.0.0.0 或 8	11111111.00000000.00000000.00000000
网络地址	10.0.0.0	00001010.00000000.00000000.00000000
广播地址	10.255.255.255	00001010.11111111.11111111.11111111

127.0.0.1/255.0.0.0 这个特殊地址会始终执行本地系统（localhost），因此它可以使用网络协议进行自我通信。

### 1) IPv4 地址分类

一开始，32 位的 IPv4 地址只由 8 位的网络地址和“剩下的”主机位组成。这种格式用在局域网出现之前。这使得独立的网络的数量不能太多（最多 254 个），在那时，只有一些很少很大的网络，例如 ARPANET。在局域网出现的早期，IPv4 地址就已经显得不够用了。

为了和已存在的 IP 地址空间及 IP 数据报兼容，对 IP 地址的定义在 1981 年的 RFC 791 中进行了修改。修改后的 IP 地址共有 5 类地址，如表 8-4 和表 8-5 所示。

表8-4 IPv4地址分类（1）

分 类	前 缀 位	开始地址	结束地址	默认子网掩码
A 类地址	0	0.0.0.0	127.255.255.255	255.0.0.0
B 类地址	10	128.0.0.0	191.255.255.255	255.255.0.0
C 类地址	110	192.0.0.0	223.255.255.255	255.255.255.0
D 类地址	1110	224.0.0.0	239.255.255.255	未定义
E 类地址	1111	240.0.0.0	255.255.255.255	未定义

表8-5 IPv4地址分类（2）

分 类	网络地址位数	剩余的位数	网 络 数	每个网络的主机数
A 类地址	8	24	128	16 777 214
B 类地址	16	16	16 384	65 534
C 类地址	24	8	2 097 152	254
D 类地址（群播）	未定义	未定义	未定义	未定义
E 类地址（保留）	未定义	未定义	未定义	未定义

可用的主机地址总数是  $2^N - 2$  ( $N$  是所用的位数, 减 2 是因为第一个和最后一个地址都是无效的)。因此, 对于用 8 位来表示主机地址的 C 类地址来说, 主机数就是 254。

更多的网络位允许更多的网络, 因此适应了互联网的持续增长。

有一些地址被设计用作私有地址。在互联网的地址架构中, 专用网络是指遵守 RFC 1918 和 RFC 4193 规范, 使用私有 IP 地址的网络。私有 IP 地址无法直接连接互联网, 需要公网 IP 地址转发。与公网 IP 地址相比, 私有 IP 地址是免费的, 也节省了 IP 地址资源, 适合在局域网使用。私有 IP 地址在 Internet 中不会被分配。私有地址的定义范围如表 8-6 所示。

表8-6 IPv4私有地址

IP 地址区段	IP 数量	分类网络说明	主机端位长
10.0.0.0 ~ 10.255.255.255	16 777 216	单个 A 类网络	24 位
172.16.0.0 ~ 172.31.255.255	1 048 576	16 个连续 B 类网络	20 位
192.168.0.0 ~ 192.168.255.255	65 536	256 个连续 C 类网络	16 位

## 2) 子网划分

在一个 IP 网络中划分子网, 可使我们能将一个单一的大型网络分成若干较小的网络。引入子网划分的概念, 可以减少因特网路由表中的表项数量 (通过隐藏一个站点内部所有独立子网的相关信息), 以及减少网络开销 (因为它将接收 IP 广播的区域划分成了若干部分)。

**示例:** 将 192.168.20.0/24 这个 C 类地址子网掩码变成 25 位, 分成两个子网, 如表 8-7 所示。

表8-7 子网划分

子网掩码	11111111.11111111.11111111.10000000 (255.255.255.128)
子网数	$2^1=2$
每个子网主机数	$2^7 - 2=126$
网络号	192.168.20.0    192.168.20.128
广播号	192.168.20.127    192.168.20.255

## 6. IPv6 地址

网际协议第 6 版 (Internet Protocol version 6, IPv6) 是网际协议的最新版本, 用于数据包交换互连网络的网络层协议, 旨在解决 IPv4 地址枯竭问题。

IPv6 地址采用了 128 位的地址, 以 16 位为一组, 每组以冒号 “:” 隔开, 可以分为 8 组, 每组以 4 位十六进制数方式表示 (0000 ~ ffff)。例如:





```
2001:0db8:85a3:08d3:1319:8a2e:0370:7344
```

由于 IPv6 地址很长，因此可以使用一些规则对它们进行简化。以下是简化规则：

- 每项数字前导的 0 可以省略，省略后前导数字仍是 0 则继续。
- 可以用双冒号 “::” 表示一组 0 或多组连续的 0，但只能出现一次。

例如，::1 是 IPv6 版本的 127.0.0.1 本地系统地址。

## 7. 网络路由和 DNS 概念

不管使用 IPv4 还是 IPv6，网络通信都需要以主机到主机和网络到网络的形式进行传输。每台主机都有一张路由表，它用于告诉主机使用哪些网络接口与主机直接连接到的子网进行通信。如果网络通信未寻找到其中的一个子网，路由表通常会有针对所有其他网络的一个条目，该条目指向可访问子网中的路由器或网关。

如果路由器收到的通信并非将其作为寻址目标，则路由器不会忽略该通信，而是根据自己的路由表转发该通信。这种处理方式可能会将通信直接发送到目标主机（如果路由器恰巧位于同一子网中），也可能转发到其他路由器。这种转发过程会一直进行，直到通信达到最终目标。

IP 协议使用地址进行通信，但是比起冗长且难记的数字字符串，操作人员宁愿使用名称。DNS（即域名系统）是分布式服务器网络，可将主机名映射为 IP 地址。为使名称服务器起作用，需要在主机系统中设置名称服务器的 IP 地址。该名称服务器无须与主机位于同一子网中，只要主机能够访问即可。

## 8.2 Linux的网络配置

在 RHEL 中，配置联网最简单的方式是使用 NetworkManager 应用程序。它可以设置可影响所有用户的系统范围内的默认值，也可以配置仅在特定用户登录时激活的特定网络接口。

### 1. 使用图形界面（GUI）工具配置主机网络连接

配置 IPv4 网络连接的步骤如下。

**Step 01** 在面板 NetworkManager（网络连接小图标）上单击鼠标右键，在弹出的快捷菜单中选择“编辑连接”命令，弹出“网络连接”对话框，如图 8-2 所示。

**Step 02** 选择一个连接设备，比如“System eth0”表示系统的第一张有线网卡，单击“编辑”按钮，打开网卡配置对话框。

**Step 03** 勾选“自动连接”和“对所有用户可用”复选框，以便立即生效和重启后提供所有用户网络连接。

**Step 04** 选择“IPv4 设置”选项卡，根据当前的网络环境，方法可选“DHCP”或“手动”。



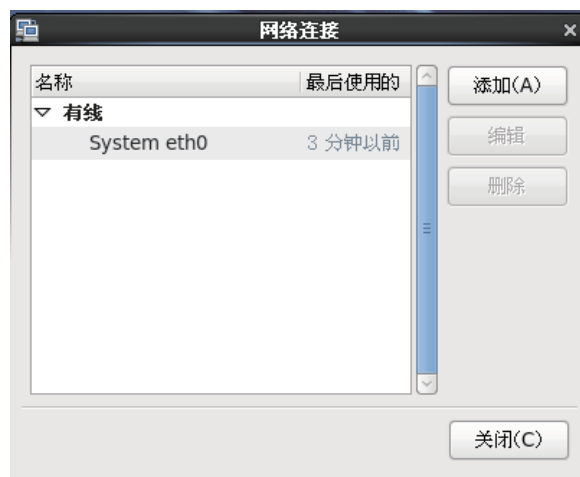


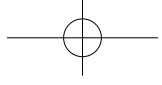
图8-2 网络连接

**Step 05** 选择“手动”时，单击“添加”按钮并指定 IPv4 地址、子网掩码、网关和 DNS 服务器 IP 地址。配置过程如图 8-3 所示。



图8-3 编辑网卡配置

**Step 06** 单击“应用”按钮，在弹出的授权对话框中输入 root 用户的密码，完成配置。

**REAL-WORK TIP**

一般来说，作为应用服务器的主机，都要配置静态 IP 地址。

在面板 NetworkManager(网络连接小图标)上单击鼠标右键，在弹出的快捷菜单中选择“连接信息”命令，可以查看当前的网络连接信息，如图 8-4 所示。如果刚刚的配置没有立即生效，可以先取消勾选“启用联网”复选框，然后再重新勾选该复选框，即可使新的配置生效。

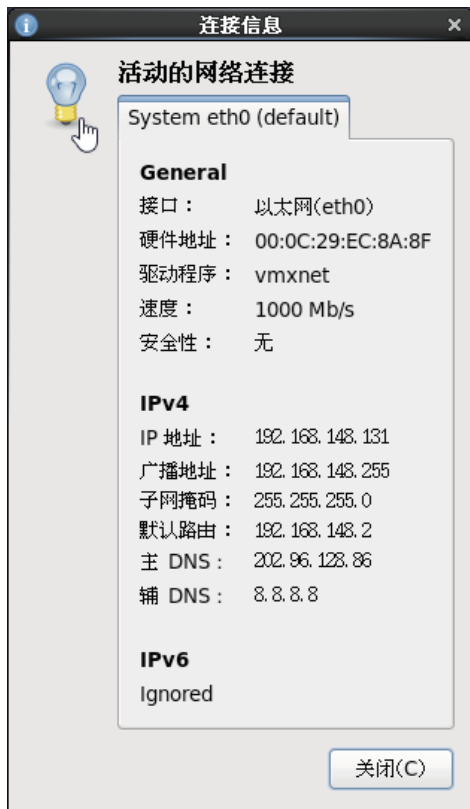


图8-4 连接信息

配置完网络连接后，应该验证配置是否正确，确定主机是否可以访问局域网中的其他主机或者访问外网。验证的方式很多，在图形界面下，可以打开 Firefox 浏览器访问互联网网站，或者使用 Nautilus 文件浏览器访问远程文件系统。

我们在这里介绍一些验证网络连接的命令。

**示例：查看 IP 地址。**

```
[root@365linux ~]# ip addr show
...
inet 192.168.148.131/24 brd 192.168.148.255 scope global eth0
```

**示例：查看路由表。**

```
[root@365linux ~]# ip route
```



```
192.168.148.0/24 dev eth0 proto kernel scope link src 192.168.148.131
metric 1
default via 192.168.148.2 dev eth0 proto static
```

该路由表显示，主机通过 **eth0** 直接路由到 192.168.148.0/24 上的主机，并且源 IP 地址为 192.168.148.131。要发送给其他网络上的主机的数据包将发送至地址为 192.168.148.2 的路由器，该路由器可以通过 **eth0** 访问。

**示例：**查询 DNS 域名解析。

```
[root@365linux ~]# nslookup www.365linux.com
Server:                202.96.128.86
Address:                202.96.128.86#53

Non-authoritative answer:
Name:                   www.365linux.com
Address:                121.9.223.176
```

**示例：**通过发送 ping 确定网络连通。

```
[root@365linux ~]# ping www.qq.com
PING www.qq.com (182.254.1.167) 56(84) bytes of data.
64 bytes from 182.254.1.167: icmp_seq=1 ttl=128 time=8.71 ms
64 bytes from 182.254.1.167: icmp_seq=2 ttl=128 time=9.79 ms
```

## 2. 使用终端界面（TUI）工具配置主机网络连接

具体操作步骤如下。

**Step 01** 切换到 root 用户，在命令行终端提示符中输入 **system-config-network-tui** 命令打开配置界面。

```
[root@365linux ~]# system-config-network-tui
```

**Step 02** 在打开的 TUI 界面上按方向下键选择“设备配置”，如图 8-5 所示，按 Enter 键确认。

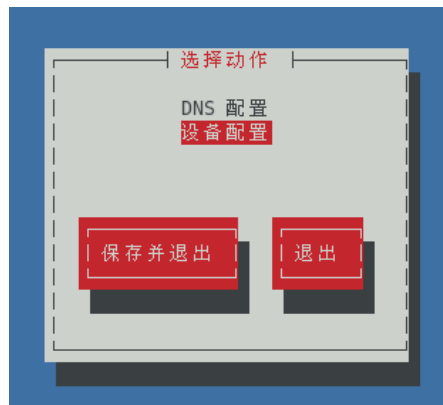


图8-5 选择动作



**Step 03** 在“选择设备”界面中选择“eth0”或者您想配置的其他网络设备，如图 8-6 所示，按 Enter 键确认。

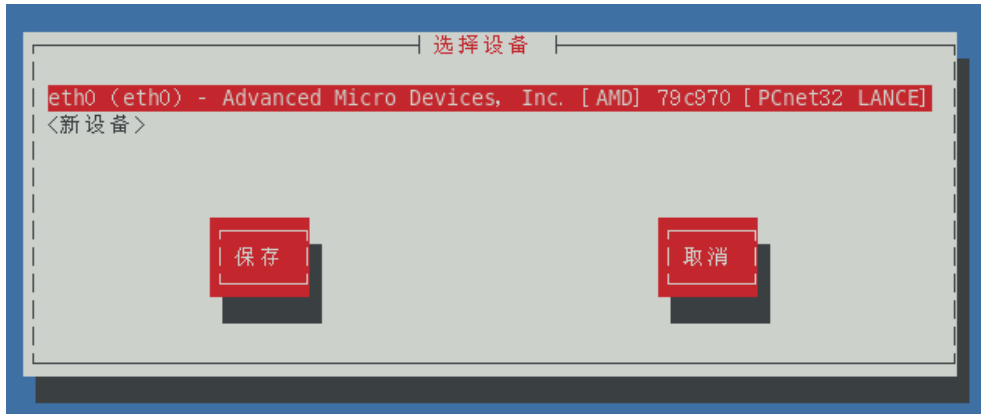


图8-6 选择设备

**Step 04** 在“网络配置”界面中填写 IP 地址等信息。按空格键可勾选（或取消），按方向上下键或 Tab 键可以移动光标位置。输入完成后，切换光标到“确定”按钮，按 Enter 键确定，如图 8-7 所示。

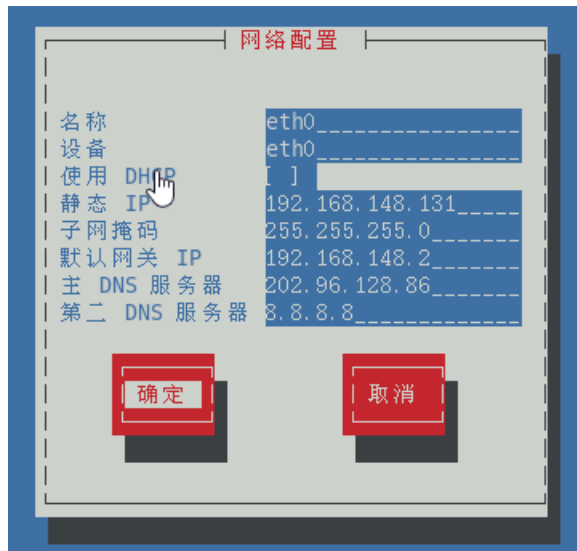


图8-7 网络配置

**Step 05** 返回“选择设备”界面，移动光标到“保存”位置上，按 Enter 键完成。

**Step 06** 再返回到“选择动作”界面，移动光标到“保存并退出”位置上，按 Enter 键完成。

**Step 07** 配置完成后网络连接如果没有立即生效，可执行命令重启 NetworkManager 服务。



```
[root@365linux ~]# service NetworkManager restart
```

**Step 08** 根据前文所述的方法自行验证网络连接是否成功。

### 3. 使用命令行配置主机网络连接

使用图形界面 NetworkManager 配置网络连接时,如果勾选了“对所有用户可用”复选框,则 /etc/sysconfig/network-scripts 中的网络配置文件将会更新,并存储配置设置。

我们也可以通过命令行 vim 工具直接编辑网卡的配置文件达到同样的效果。

**示例:** 使用 vim 编辑 eth0 的配置文件。

```
[root@365linux ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=192.168.148.131
PREFIX=24
GATEWAY=192.168.148.2
DNS1=202.96.128.66
DNS2=8.8.8.8
```

以上仅列出必不可少的关键项。



#### NOTE

如果 NetworkManager 服务正在运行,对 ifcfg-\* 所做的任何更改将立即生效。

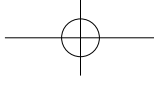


#### REAL-WORK TIP

NetworkManager 在 RHEL 6 系统中默认情况下会自动开机运行,并自动根据设置管理主机各个设备的网络连接。但 NetworkManager 可能不支持一些高级的网络应用,比如双网卡绑定、桥接网络等。这种情况下,需要添加“NM\_CONTROLLED=no”到网卡的配置文件中,永久性地手动管理网络设置。

## 8.3 远程管理Linux系统初探

当 Linux 主机建立好网络连接后,我们便可以远程从网络访问和管理系统。SSH 是最通用的远程系统管理工具之一。它允许用户登录远程系统并在其上执行命令。SSH 使用加密技术在网络中传输数据,具有很高的安全性。在绝大多数 Linux 系统中,SSH 是标准配置,并且默认启动,并允许访问(防火墙默认开放相应端口)。



本节我们简单介绍 RHEL 系统中 OpenSSH 的应用，方便大家在学习过程中使用。关于 OpenSSH 服务的配置和其他 Linux 系统远程管理的工具将在后面介绍。

RHEL 系统中 SSH 服务默认启动，并且无须配置即可使用。将 RHEL 主机作为服务器端，在网络连接畅通的情况下，可以使用 SSH 客户端工具很方便地连接到 RHEL 主机。

根据不同的系统环境，有多种 SSH 客户端软件，而且大多数是开源软件。

- Linux 系统自带客户端。
- Windows 系统需要下载（安装）SSH 客户端。

## 1. Linux 系统下的 SSH 客户端

Linux 系统下的 SSH 客户端即默认自带的 ssh 命令。

示例：使用 ssh 命令连接 Linux 主机。

```
[demo@localhost ~]$ ssh zhangsan@192.168.148.131
The authenticity of host '192.168.148.131 (192.168.148.131)' can't be
established.
RSA key fingerprint is 2b:86:16:de:86:31:91:0d:ae:0a:a3:bd:59:23:06:76.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.148.131' (RSA) to the list of known
hosts.
reverse mapping checking getaddrinfo for bogon [192.168.148.131] failed -
POSSIBLE BREAK-IN ATTEMPT!
zhangsan@192.168.148.131's password:
[zhangsan@365linux ~]$
```

以上命令使用 zhangsan 用户身份连接登录 192.168.148.131 这台主机。zhangsan 是远程服务器上的用户。如果在连接时不指定用户（即 ssh 命令后面直接跟 IP 地址），则 ssh 命令使用当前执行命令的用户身份尝试登录远程主机，这需要远程主机上必须存在和本地客户端主机相同的用户。

第一次建立连接时，需要进行身份确认。在询问是否继续时输入 yes。

输入远程主机将登录用户的密码，即可成功登录。

登录成功后，当前终端环境变成远程主机的终端环境。

示例：退出远程主机登录。

```
[zhangsan@365linux ~]$ exit
logout
Connection to 192.168.148.131 closed.
```

## 2. Windows 系统下的 SSH 客户端

Windows 系统下 SSH 客户端推荐如下。

- PuTTY：开源软件，免费使用，软件小巧，免安装，方便携带。
- Xshell：商业软件，对学校、家庭使用免费，功能强大。

**REAL-WORK TIP**

所有使用的服务器远程管理客户端软件，首先选用开源软件，且软件一定要在官方网站上（建议使用 Google 进行搜索）下载原版（禁止使用所谓的汉化版或者破解版）。尽可能对下载的软件进行 md5/sha1 签名验证。一旦使用来路不明可能含有木马后门的 SSH 客户端软件，将对服务器系统造成巨大的安全威胁。

**示例：**使用 PuTTY 远程连接 Linux 系统。

**Step 01** 到 PuTTY 官方网站 <http://www.putty.org/> 下载 PuTTY 软件，如图 8-8 所示，并进行签名验证。

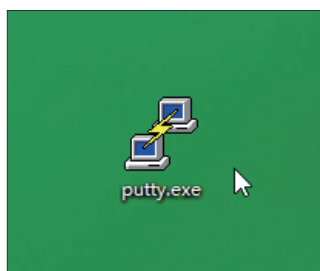


图8-8 PuTTY软件图标

**Step 02** 无须安装，直接运行。填写 IP 地址和端口号（默认为 22），单击“Open”按钮，如图 8-9 所示。

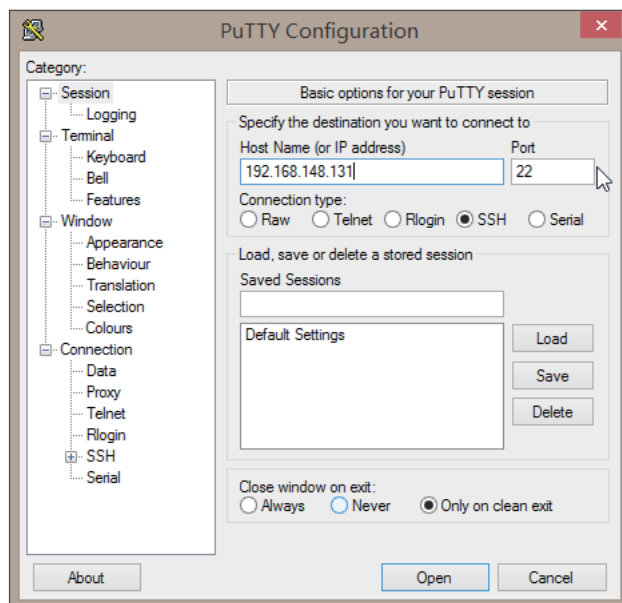
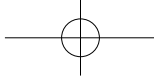


图8-9 启动PuTTY



**Step 03** 输入要登录的用户名，这里用 root 用户登录；然后输入用户密码（输入密码过程中，密码不回显在屏幕上），如图 8-10 所示，按 Enter 键确定。



图8-10 PuTTY登录过程

**Step 04** 登录成功，效果如图 8-11 所示。

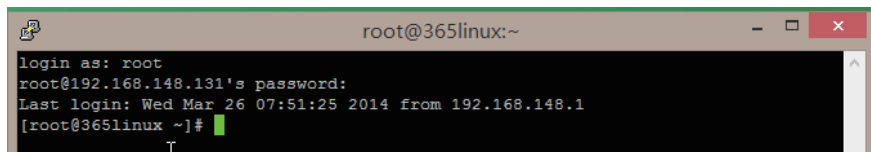


图8-11 登录成功



### REAL-WORK TIP

在生产环境系统中，为安全起见，SSH 服务器一般配置为禁止 root 用户远程登录。正确的做法是使用一个普通用户登录到系统，然后需要 root 权限时再使用 su 或者 sudo 切换到管理员。



## 本章总结

Linux 系统广泛应用于服务器领域，每一个服务的提供都依赖于网络通信，所以为系统建立网络连接是对系统进行配置的第一步。对于单台主机的网络配置相对简单，而当服务器融入到复杂的生产环境的网络中时，对于主机网络的配置，更多的是要考虑网络的规划及如何在复杂的网络中进行合理的连接，以在实现网络通信时保证网络的安全。这些都需要基本





(甚至深入) 的网络相关知识作为支持。对于非专职从事网络管理的系统工程师而言, 至少要理解网络架构的设计及主机在该网络中所处的位置, 当出现网络通信问题时, 能够判断故障是发生在网络层面还是系统层面。

### 章节重点:

- IP 地址、子网划分、路由及 DNS 的概念。
- 使用图形界面工具配置系统 IP 地址。
- 使用命令行工具配置系统 IP 地址。
- 使用 SSH 客户端工具连接到系统。



## 基础技能演练

1. 关于建立网络连接, 回答以下问题:
  - (1) 目前你知道几种方式查看系统的 IP 地址?
  - (2) 使用什么命令可以简单地测试两台主机之间是否有网络连接?
  - (3) 网关的作用是什么?
  - (4) DNS 的作用是什么?
2. 完成任务 (具体的 IP 地址请根据真实环境或讲师的要求进行设计):
  - (1) 使用图形工具配置主机的 IP 地址、网关和 DNS。
  - (2) 使用命令行工具配置主机的 IP 地址、网关和 DNS。
  - (3) 使用命令行查看和验证网络连接信息。
3. SSH 客户端的使用。
  - (1) 在配置好网络连接后, 在两台 Linux 主机之间使用 ssh 命令进行远程登录。
  - (2) 使用 Windows 下的 PuTTY 软件远程登录到 Linux 主机。



## 提高技能实践

1. 如果 Linux 主机 A 无法 ping 通 Linux 主机 B, 根据已有的知识, 分析一下可能有哪些方面的原因。
2. 关于子网划分, 回答以下问题:
  - (1) 172.16.46.255/20 是一台合法主机吗?
  - (2) 路由器在其接口上收到一个目标地址是 172.16.46.191/26 的数据包, 则路由器会怎样做?
  - (3) 有一家公司有五个部门, 分别是财务 7 人, 技术 15 人, 人事 6 人, 销售 17 人, 管理 10 人, 要求每个部门一个子网, 使用 C 类地址, 应该怎样划分最合理?



## 第9章

# 管理物理存储和建立文件系统

在安装 Linux 系统时，根据服务器的用途，需要对硬盘进行合理的分区规划。在系统运行的过程中，随着业务的发展变化，以及不断增长的数据存储的要求，也需要对磁盘存储进行调整和管理。目前在服务器上使用得最多、技术最成熟可靠的仍然是传统的机械硬盘。随着技术的发展，在相当多的领域也使用了基于闪存技术的 SSD 硬盘。在这一章节，对于硬盘的物理组成介绍主要指的是机械硬盘。

### 9.1 硬盘的物理组成

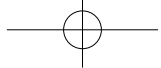
硬盘是由多个盘片组成的，按照硬盘片能够容纳的数据量，分为单盘（一块硬盘里面只有一个硬盘片）和多盘（一块硬盘里面有多块硬盘片）。

硬盘里有磁头（Head）在磁盘上读 / 写，磁头固定在机械手臂上，机械手臂上有多个磁头，可以进行读 / 写。当磁头固定不动时，硬盘片转一圈所画出来的圆就是磁道（Track）。如前所述，一块硬盘可能有多个盘片，所有硬盘上相同半径的磁道就组成了柱面（Cylinder）。在 RHEL 5 的版本中，柱面是分区时的最小计量单位。

由圆心向外画直线，可以将磁道再细分为扇区，扇区就是硬盘上的最小存储物理量。通常，一个扇区的大小约为 512 字节。而在 RHEL 6 的版本中，分区的最小计量单位是扇区。

当硬盘读取数据时，硬盘会转动，利用机械手臂将磁头单方向地来回移动定位到正确的数据位置，将数据准确读出。硬盘的磁头与硬盘片的接触空间很细微，如果有抖动或者是有脏污，就会造成数据或物理硬盘的损坏。所以在使用计算机的过程中，不要移动主机，以避免硬盘抖动。关机时使用正确的关机方式而不是直接关闭电源，因为正确关机过程中，系统会使硬盘的机械手臂回归原位。

目前新兴的固态硬盘在物理构造上与传统的机械硬盘完全不同，它用电子芯片存储取代



了磁介质存储，因此在 SSD 固态硬盘中没有磁头的机械运动和盘片的转动。

## 9.2 分区和文件系统的基本概念

在开始对硬盘分区之前，必须要了解一下根分区和文件系统相关的一些基本概念。

### 1. MBR

主引导记录（Master Boot Record，MBR）是计算机系统开机后访问硬盘时所必须要读取的首个扇区，它在硬盘上的三维地址为 0 柱面 0 磁头 1 扇区。所以 MBR 的容量大小就是一个扇区的大小，即 512 字节。而其中开头的 446 字节内容为系统的“主引导记录”（MBR）信息，为避免混淆，称为 BootLoader；其后的 64 字节被分成 4 个 16 字节的“磁盘分区表”（DPT），即 Disk Partiton Table；以及最后 2 字节的结束标志（55AA）。

MBR 的组成如图 9-1 所示。

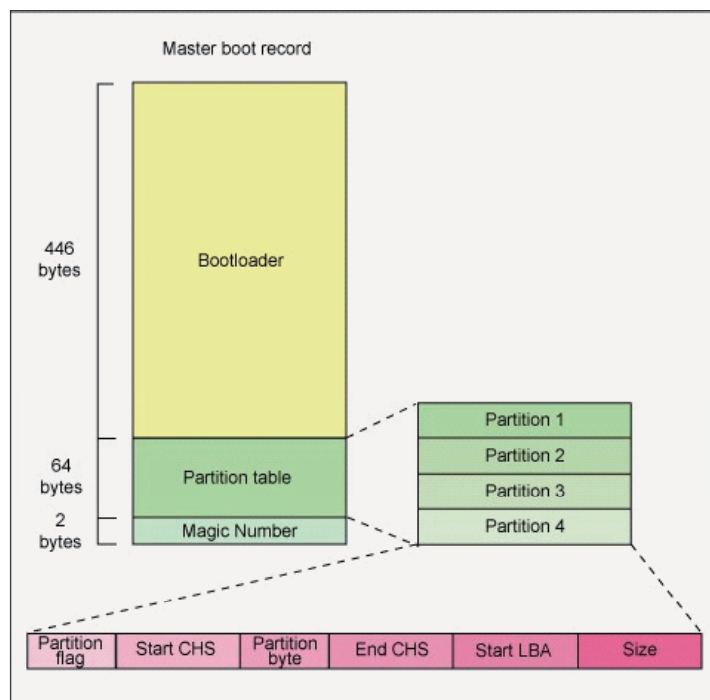


图9-1 MBR的组成

很显然，跟硬盘分区有关的就是 Partition table，它的 4 个 16 字节的固定设计决定了一个硬盘最多只能够分出 4 个主分区。如果需要 5 个以上的分区，那就需要将其中一个主分区变为扩展分区，在扩展分区的基础上再建立逻辑分区。



## 2. 文件系统

硬盘当然是用来存储数据的，即存放系统和用户的文件，而分区的文件系统就规定了数据存储单元在硬盘上排列的规则。不同的文件系统，存储数据的方式不一样，这些不同的文件系统可以共存于同一个系统的硬盘之上，但不能共存于同一个磁盘分区之上。Linux 最重要的特性之一就是支持多种文件系统。而 Linux 通过 VFS（Virtual File System）支持多个不同的文件系统，能够与不同的操作环境实现资源共享。

Linux 支持的文件系统包括：Ext2、Minix、MS-DOS、FAT、ISO9660、Ext3、Ext4、ReiserFS、NTFS、JFS、XFS、NFS、SMBFS。

**示例：**可以通过以下命令查看当前 Linux 系统支持的文件系统。

```
[root@365linux ~]# ls -l /lib/modules/`uname -r`/kernel/fs
```

**示例：**当前系统启用的文件系统。

```
[root@365linux ~]# cat /proc/filesystems
```

默认情况下，RHEL 6 采用 Ext4 文件系统。

## 9.3 建立和使用磁盘分区

在 RHEL 6 系统中，使用 fdisk 命令进行分区操作。

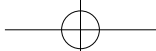
### Step 01 查看现有的系统磁盘分区。

```
[root@365linux ~]# fdisk -cul
```

```
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000caf05

Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048     1026047       512000   83   Linux
/dev/sda2             1026048     9414655      4194304   82   Linux swap / Solaris
/dev/sda3             9414656    41943039     16264192   83   Linux

Disk /dev/sdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```



查看整个系统内的设备分区列表。通过分析列出的信息，我们可以了解到该硬盘的使用情况。如上例中所示，该系统存在两块硬盘，名称分别为 `sda` 和 `sdb`，其中 `sda` 是在安装系统时进行了规划，从命令输出的结果数据表明，它的容量大小为 21.5GB，共有 41 943 040 个扇区；`sdb` 的容量大小是 10.7 GB，共有 20 971 520 个扇区。

`sda` 被分成 3 个分区：`sda1`、`sda2`、`sda3`，从分区的起始扇区可以看到 3 个分区连续排列，且最后一个分区 `sda3` 的最后结束扇区为 41 943 039，接近于硬盘的扇区总数，说明该硬盘已没有剩余的空闲空间可以进行分区操作。

`sdb` 则是一个没有任何分区的、完整的、空闲的硬盘。接下来使用 `fdisk` 命令对 `sdb` 进行分区的操作。

#### Step 02 分区操作。使用 `fdisk` 命令进入交互分区模式。

```
[root@365linux ~]# fdisk -cu /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0x1dfc88be.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by
w(rite)

Command (m for help): n          // 输入 n 创建新的分区，可以通过输入 m 获得命令的帮助
Command action
e   extended
p   primary partition (1-4)
p   // 输入 p 选择创建主分区
Partition number (1-4): 1        // 分区编号为 1
First sector (2048-20971519, default 2048):          // 起始扇区，不填默认为 2048
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +2G
// 分区结束扇区，可以使用扇区数，也可以使用单位大小，比如 +2G 即表示分 2GB 大小的分区，比较直观

Command (m for help): p          // 分完区后输出当前的分区表查看 sdb 的分区情况

Disk /dev/sdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1dfc88be

Device Boot      Start         End      Blocks      Id      System
/dev/sdb1        2048      4196351      2097152      83      Linux
```



```
Command (m for help): w          // 输入 w 保存分区表到 MBR 的第二区域 DPT
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

使用 `fdisk` 对分区进行新建、修改、删除的操作，在输入 `w` 保存命令之前都是没有写入分区表的，随时可以撤销，或者输入 `q` 不保存退出。

**NOTE**

`fdisk` 命令工具创建的 DOS 格式的分区表，是一种最常见而且最大程度兼容其他操作系统的分区表，但因为历史的原因 DOS 格式的分区表最大只能支持 2TB 的分区大小。

**REAL-WORK TIP**

在生产环境中，通常情况下不允许对正在使用的硬盘进行分区操作，比如对已经安装并正在运行的操作系统的硬盘进行分区是不允许的。即使在特殊的情况下必须对正在使用的硬盘进行分区，分区调整后保存到分区表，这时正在运行的系统内核并不会识别新的分区表，也就是说调整后的分区不会被系统识别。必须进行重启或者执行 `partprobe` 或者 `partx` 强制使新的分区表生效。

**Step 03** 创建文件系统。

示例：将 `/dev/sdb1` 格式化为 Ext4 的文件系统。

```
[root@365linux ~]# mkfs.ext4 /dev/sdb1
```

**Step 04** 挂载和使用磁盘分区。

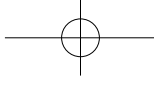
示例：将 `/dev/sdb1` 挂载自定义的 `/data` 目录。

```
[root@365linux ~]# mkdir /data
[root@365linux ~]# mount /dev/sdb1 /data
```

要使用 Linux 分区，在格式化完后，还需要对分区进行挂载（Mount）。所谓挂载，就是将需要使用的分区载入到某一个目录，该目录就称为挂载点。挂载点必须要提前存在，挂载成功后，在挂载点目录下创建的所有文件实际上就是存储在挂载到该目录下对应的分区上。

**NOTE**

若挂载点本身非空，载入分区后，原目录下的文件会暂时“消失”（不可访问），被挂载上来的分区中的内容所取代，等到该分区被卸载后，原目录中的文件就会再次显示。一般不会使用非空的目录作为常用的挂载点。



示例：查看分区挂载情况。

```
[root@365linux ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        16G   626M   14G   5%  /
tmpfs            947M     0   947M   0%  /dev/shm
/dev/sda1        477M    23M   429M   6%  /boot
/dev/sdb1        2.0G   3.0M   1.9G   1%  /data
```

使用 **mount** 命令挂载分区到挂载点，立即生效，但是系统重启后不会自动挂载。如果需要在系统每次开机时都自动将指定的分区挂载到某个特定的目录，则需要将挂载信息写入系统的文件系统列表配置文件。

示例：**/dev/sdb1** 开机自动挂载到 **/data** 目录。

```
[root@365linux ~]# vim /etc/fstab
...
/dev/sdb1          /data              ext4              defaults          0 0
```

在 **fstab** 配置文件最后添加一行挂载记录。**fstab** 文件格式分为 6 列，每一列的内容含义如下：

- 设备名或者设备的卷标。
- 挂载点。
- 分区的文件系统类型。
- 文件系统的挂载参数。
- 是否使用 **dump** 命令备份，0 表示不要做 **dump** 备份；1 表示进行 **dump** 备份；2 表示备份，但重要性比 1 小。
- 是否使用 **fsck** 命令检测，0 表示不要检验；1 是要检验；2 是要检验，不过优先级小于 1。



#### REAL-WORK TIP

在 **fstab** 中分区的设备名称使用 **/dev/sdb1** 是不推荐的，因为如果以后对该分区所在的硬盘进行分区调整，其分区号数字可能会发生变化。所以更推荐使用 **LABLE** 或者 **UUID** 的格式。

#### Step 05 卸载分区。

如果是临时挂载使用某个分区，在使用完成后，可以将其卸载。卸载之后，分区上的数据将无法访问。

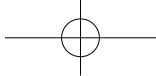
示例：通过分区设备名称卸载。

```
[root@365linux ~]# umount /dev/sdb1
```

示例：通过挂载点卸载。

```
[root@365linux ~]# umount /data
```



**NOTE**

特别注意的是，在卸载分区之前，先要确保该分区数据未被占用使用，包括当前用户退出其挂载点目录。

**本章总结**

在 Linux 系统管理的工作中，对于存储设备的管理是日常操作之一。在本章节我们了解了磁盘存储的物理结构构成，理解了分区的实质存在、分区的意义和具体操作。由于 Linux 的基于文件的设计理念，所以对于分区的访问，是需要通过一个目录作为分区对应的入口，这个目录被称为挂载点。这一点必须要深刻理解，在工作中，灵活运用这样的特性，可以发掘出实际使用的很多技巧，比如数据的迁移。

Linux 对多种文件系统都有很好的支持，而且在使用上保持了一致性，限于篇幅和难度设计，这里并没有展开讨论。例如，对于 FAT 或 NTFS 文件系统的支持，对于远程文件系统的挂载使用，SWAP 交换分区的调整等。

**章节重点：**

- 理解分区和文件系统的基本概念。
- 对磁盘进行分区、格式化、挂载和卸载。

**基础技能演练**

要求创建一个 Ext4 格式的分区，大小为 1GB，ID 指示为 linux。格式化后，挂载到 /media/test，并在 /media/test/ 目录内创建一个文件 mount.txt。然后将其设为开机自动挂载，重启检查实验结果。

**提高技能实践**

1. 要求创建新的一个分区，大小为 512MB，ID 指示为 linux swap。
  - (1) 将该 512MB 的分区设置成交换分区，并启用，用 free 查看实验结果。
  - (2) 再使用 dd 命令创建一个 512MB 的交换文件，设置成交换文件，并启用。
2. 如果 /home 目录并不是一个单独的分区，请在现有的系统中新建一个 2GB 的分区，将 /home 目录移动到该分区。移动完成后，不影响整个系统和 /home 目录的文件结构与用户使用。





## 第10章

# 逻辑卷的使用

在生产环境中，随着业务的增长，企业在服务器上需要存储的数据越来越多，需要更大的存储空间，最初安装系统时规划的分区大小已无法满足数据存储需求，怎么办？

LVM（Logical Volume Manager）即逻辑卷管理器，是一种强大的卷管理工具。LVM 通过将一个或者几个物理分区组合在一起，整合成一个磁盘分区的集合，然后在这个集合中划分出独立的逻辑卷，而逻辑卷看起来就如同一个大的磁盘分区。这样的整合方式可以实现硬盘空间的动态划分和管理。

## 10.1 逻辑卷的概念

### 1. 物理卷

物理卷（Physical Volume, PV）处于逻辑卷管理器中的最底层，可以是磁盘分区，也可以是整个磁盘。只有将分区或磁盘变成 PV 后，才能建立 LVM。

### 2. 卷组

卷组（Volume Group, VG）建立在物理卷之上，相当于一个磁盘容器。一个卷组中包括一个或多个物理卷，并可以在卷组建立之后动态添加物理卷到卷组中。一个系统中可以有多个 VG。

### 3. 逻辑卷

逻辑卷（Logical Volume, LV）建立在卷组之上，是最后挂载到系统的“分区”。逻辑卷建立后可以动态扩展或缩小空间。一个 VG 中可以有多个 LV。

### 4. 物理扩展

物理扩展（Physical Extend, PE），在建立 VG 的时候，可以指定 PE 数值，默认是 4MB，这是 VG 中可以用于分配的最小存储单位。



## 10.2 配置LVM的步骤

示例：在系统中已经准备好 3 个分区，分别为 /dev/sdb5、/dev/sdb6、/dev/sdb7，大小都为 500MB，使用这 3 个分区整合逻辑卷。为了明确标识，修改磁盘 ID 为 8e，System 显示为 Linux LVM。过程略。

```
[root@365linux ~]# fdisk -l |grep LVM
/dev/sdb5          245          306          497983+   8e   Linux LVM
/dev/sdb6          307          368          497983+   8e   Linux LVM
/dev/sdb7          369          430          497983+   8e   Linux LVM
```

### Step 01 生成物理卷 (PV)。

```
[root@365linux ~]# pvcreate /dev/sdb{5,6}
```

将分区 /dev/sdb5、/dev/sdb6 生成物理卷。注：sdb7 暂不使用。

```
[root@365linux ~]# pvdisplay
```

查看系统中 PV 的信息。

### Step 02 创建卷组 (VG)。

```
[root@365linux ~]# vgcreate vg001 /dev/sdb5 /dev/sdb6
```

创建一个新的卷组 vg001，卷组包括两个物理卷 sdb5 和 sdb6。

```
[root@365linux ~]# vgdisplay
```

查看系统中 VG 的信息。

### Step 03 划分逻辑卷 (LV)。

```
[root@365linux ~]# lvcreate -L 500m -n data vg001
```

从卷组 vg001 中划分 500MB 空间创建一个逻辑卷，命名为 data。可以使用 -l n 来定义逻辑卷空间大小，大小为 PE Size 乘以 n。n 的取值取决于 Free PE 的大小。

```
[root@365linux ~]# lvdisplay
```

至此，LVM 就已经配置完毕。现在将 LV 格式化后，就可以挂载使用了。

示例：格式化逻辑卷，并将其挂载到 /data 目录。

```
[root@365linux ~]# mkfs.ext4 /dev/vg001/data
[root@365linux ~]# mount /dev/vg001/data /data
[root@365linux ~]# df -h |grep data
/dev/mapper/vg001-data    485M    11M   449M    3% /data
```

因为一个 LV 可能是由多个分区，甚至多个不同硬盘的空间组成的，这种情况下当数据写入 LV 时，实际上是存储到多个不同的设备上。LVM 支持两种不同的写入机制。

(1) 线性模式 (linear)：如果 LV 中包括了两个磁盘分区 /dev/sdb5 和 /dev/sdb6，则线性模式在写入数据时，先写满 sdb5 后才会使用 sdb6。LVM 在建立时，默认使用线性模式。

(2) 条带模式 (stripe)：条带模式在写入数据时，会拆分后分别写入到两块磁盘，理论上，读 / 写性能会更好。



通常, LVM 最主要的用途是管理一个大的存储空间, 方便灵活地调整分区的大小。使用线性模式, 有利于增加或减少逻辑卷的大小。如果想使用条带模式的高性能特性, 一般会采用 RAID 技术来管理磁盘 (本教程不涉及 RAID 内容)。

## 10.3 扩大逻辑卷容量

当 LV 用作数据存储被使用时, 如果容量不够, 可以很方便地进行容量的扩大。因为 LV 是从 VG 中分配的, 在扩大 LV 容量时, 先看看该 LV 所属的卷组中还有没有剩余的空间 (Free PE)。可以通过 `vgdisplay` 命令查看。

**示例:** 当 VG 中存在 Free PE 时, 增加 LV 容量。

```
[root@365linux ~]# lvextend -l +117 /dev/vg001/data
```

从 `vg001` 中划分 117 个 PE 增加逻辑卷 `data` 的容量。

```
[root@365linux ~]# resize2fs /dev/vg001/data
```

在线扩展正在使用的文件系统的容量。

**示例:** 当 VG 没有 Free PE 可用时, 就需要先增加 VG 的容量。

```
[root@365linux ~]# pvcreate /dev/sdb7
```

新增一个物理卷。

```
[root@365linux ~]# vgextend vg001 /dev/sdb7
```

将新增的物理卷添加到卷组 `vg001` 中。

VG 增大后, 使用 `lvextend` 命令增大 LV 的容量。



### REAL-WORK TIP

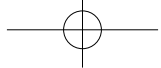
在实际生产环境中, 对于 LVM 逻辑卷的容量调整一般是增大容量, 很少有需要减少的情况。如果需要减少, 需要先停止使用 LV, 并考虑减少量和数据大小的匹配。



## 本章总结

在企业的实际应用中, 对于存储的管理, LVM 应用得非常多, 因为它功能强大、管理灵活。RHEL 6 在安装系统时, 默认的分区方式就是使用 LVM 管理分区。使用 LVM 逻辑卷, 除了方便在线动态调整逻辑卷使用容量外, 还有虚拟容量、磁盘快照等高级功能。

LVM 的管理命令非常多, 且功能复杂, 在这里不可能一一解释, 我们要在理解 LVM 的基础上, 借助帮助文档, 或阅读相关的技术文档, 加深对 LVM 管理命令的理解。在实际应用中, 碰到任何问题, 都能举一反三, 找到合适的解决方法。



### 章节重点：

- 理解 LVM 的原理和使用场景。
- LVM 创建和在线扩容。



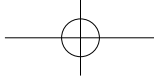
## 基础技能演练

1. 创建逻辑卷。
  - (1) 使用两块虚拟硬盘，一块 2GB，一块 5GB，组成卷组 `vg_test`。
  - (2) 从卷组 `vg_test` 中划分逻辑卷 `lv_test`，大小为 4GB。
  - (3) 格式化 LV 为 Ext4 文件系统，并挂载到 `/test` 目录。
  - (4) 使用 `dd` 命令在 `/test` 目录中创建一个大小为 3GB 的文件。
2. 增加逻辑卷的容量。
  - (1) 将逻辑卷 `lv_test` 的容量在线调整到 6GB。
  - (2) 将逻辑卷 `lv_test` 的容量在线调整到 8GB。



## 提高技能实践

某服务器上一个正在使用的逻辑卷是由位于 3 块不同硬盘上的分区组成的，假设逻辑卷的数据大小只占用不到一块硬盘分区的容量，现在要在保证数据安全并不影响 LVM 结构的情况下，减小该逻辑卷的容量，并退出其中一块硬盘以做其他用。可以实现吗，如何操作？



## 第 11 章

# 安装和管理软件包

在使用桌面操作系统时，因为工作和生活的需要，需要在系统中安装各种软件包。同样的 Linux 操作系统不管是作为桌面端使用，还是作为服务器系统用于生产环境，都有大量的软件包可以选择性安装。需要理解的是，不同操作系统平台的软件安装包一般是不能通用的。Linux 由于开源特性，操作系统版本众多，不同版本的操作系统软件包的管理方式各不相同。

### 11.1 软件包管理介绍

#### 1. RPM

RPM 是 Red Hat 软件包管理器。RPM 包是一种预先已经编译好的二进制文件，安装 RPM 包的优点在于安装方便，不需要重新编译源代码；RPM 在安装的过程中会记录软件包的安装信息到 RPM 数据库，方便查询、升级和卸载。但同时也带来一定的缺点，比如安装环境要与打包时的环境保持一致、需要满足安装套件的依赖关系等。

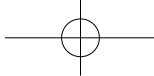
使用 RPM 安装软件包，一般会执行如下操作。

- (1) 检查软件包的依赖关系和软件包冲突。
- (2) 执行安装前的脚本程序。
- (3) 备份原来的配置文件。
- (4) 解压并安装软件包文件。
- (5) 执行安装后脚本程序。
- (6) 更新 RPM 数据库。
- (7) 执行安装时触发脚本程序。

一个典型的 RPM 软件包的文件名：

```
gftp-2.0.18-3.2.2.i386.rpm
```

软件包名 - 版本号 - 子版本号或发布次数 . 支持的平台 . 扩展名



## 2. SRPM

Source RPM (SRPM) 是 rpm 源码包, 通常是以 `src.rpm` 的扩展名命名。SRPM 包含有安装参数配置文件, 安装时 RPM 软件管理程序会将该源码包编译生成一个 rpm 的二进制文件, 然后再将这个 RPM 包安装进系统。SRPM 的优点在于可以通过修改参数配置文件, 重新编译产生适合当前 Linux 系统环境的 RPM 包。

## 3. YUM

RPM 在安装过程中最大的问题就是要解决依赖关系, YUM 的出现克服了这一问题。YUM 是一种基于 RPM 的智能交互式的软件管理机制。使用 YUM 安装软件时, 会通过分析 RPM 软件包的表头数据, 从 YUM Server 上下载文件进行安装。

YUM 功能强大, 可以用来安装、查询、升级和卸载软件包, 还能更新系统, 甚至搭建本地的 YUM 软件仓库。

## 4. RHN

RHN (Red Hat Network) 是一种付费的商业服务, 使用在线的软件安装和系统升级, 提供比 YUM 更多的功能和自动化能力。

## 5. DPKG

DPKG 是 Debian 的软件包安装管理工具。它和 RPM 一样功能强大。软件包一般是以 deb 包的形式发布。

## 6. APT

APT (Advanced Package Tool) 是一种和 YUM 类似的在线的软件安装管理机制, 功能强大, 而且免费。APT 源自 Debian, 但是它已经扩展到可以支持 RPM。

## 7. Tarball

Tarball 是软件源码打包发布的一种形式, 在安装的时候需要参考它提供的安装说明文档进行编译安装。其优点是灵活性强, 定制自由度高; 缺点是安装过程稍烦琐。

在本章中主要是对 RHEL 系列的 rpm 包管理的介绍。

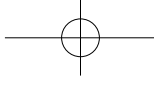
# 11.2 使用rpm命令安装和管理RPM包

rpm 命令可用来查询、安装、升级、卸载 RPM 软件包。

**示例:** 使用 rpm 命令安装软件包。

```
[root@365linux ~]# rpm -ivh packagename.i386.rpm
```

安装软件包, -i 表示 install, -v 显示安装信息, -h 显示安装进度。



示例：使用 rpm 命令在线安装软件包。

```
[root@365linux ~]# rpm -ivh http://www.51365linux.com/source/packageName.i386.rpm
```

示例：使用 rpm 命令时使用指定参数。

```
[root@365linux ~]# rpm -ivh packageName.i386.rpm --replacepks
```

重新安装软件包。

常用的参数还有如下几个。

- --nodeps：不检查软件包的依赖关系。
- --root directory：安装软件包以指定的目录作为根树。
- --force：强制覆盖安装软件包。
- --replacefiles：替换已存在的文件。
- --prefix newpath：将可重置的软件包安装到指定的路径下。
- --test：测试安装。

示例：使用 rpm 命令卸载软件包。

```
[root@365linux ~]# rpm -e packageName
```

卸载软件包时，不需要软件包的版本号，指定要卸载的软件包名即可。

示例：使用 rpm 命令升级软件包。

```
[root@365linux ~]# rpm -Uvh packageName.i386.rpm
```

升级软件包。使用 -U 选项，RPM 将自动删除系统中的旧包，但不管系统中有无旧包版本，都会进行安装。使用 -F 选项，如果系统没有该软件包的旧版本，就不会执行安装。

示例：使用 rpm 命令查询某个软件包是否安装。

```
[root@365linux ~]# rpm -q packageName
```

当无法提供完整的软件包名称时，常使用 -a 参数先查询所有已安装的软件，再通过管道过滤出结果。

```
[root@365linux ~]# rpm -qa |grep packageName
```

示例：使用 rpm 命令查询某个文件是由哪个软件包安装提供的。

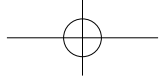
```
[root@365linux ~]# rpm -qf filename
```

常见的查询选项还有如下几个。

- -qi：列出软件包的详细信息。
- -ql：列出软件包安装的所有文件。
- -qc：列出软件包所有的配置文件。
- -qd：列出软件包所有的文档。
- -qR：列出软件包的依赖文件。

示例：使用 rpm 命令查询一个未被安装的软件包结构。

```
[root@365linux ~]# rpm -qpl packageName.i386.rpm
```



加上 **-p** 参数，表示查询的是一个未被安装到系统的软件包。

**示例：**使用 RPM 包进行文件信息校验。

```
[root@365linux ~]# rpm -V packagename
```

检查软件包所安装的文件有没有被改动过。如果没有信息出现，则表示没有被改动。如果出现以下信息，则表示文件被改动过。

- S：文件大小的改变。
- M：文件类型或属性改变。
- 5：md5 值改变。
- D：设备的 major/minor 改变。
- L：连接属性改变。
- U：文件拥有者改变。
- G：文件所属组改变。
- T：文件的建立时间改变。

**示例：**列出当前系统上所有可能被改动过的文件。

```
[root@365linux ~]# rpm -Va
```

**示例：**安装软件包的数字签名公钥。

```
[root@365linux ~]# rpm -import /media/RPM-GPG-KEY-redhat-release
```

**示例：**检查软件包的数字签名。

```
[root@365linux ~]# rpm -K packagename.i386.rpm
```

可使用 **-Kvv** 显示更详细的信息。

**示例：**使用 rpm 命令管理 RPM 软件包数据库信息。

```
[root@365linux ~]# rpm --initdb
```

如果 RPM 数据库不存在，则重新创建一个新的（而存在的不会被覆盖）。

```
[root@365linux ~]# rpm --rebuild
```

从已安装软件包的表头信息重建 RPM 数据库索引。

## 11.3 使用YUM工具安装和管理RPM包

基于 Linux 软件的开发方式和原则，每个软件提供较单一的功能，多个软件包综合完成一个大而全的任务。所以在安装 RPM 包时，经常会出现用户想要安装 A 软件包时，被提示依赖 B 软件包，甚至依赖多个其他软件包。在这种情况下，需要先安装被依赖的软件包后才能安装 A 软件包。而手动去解决依赖关系就是要找到一个个被依赖的包，这个过程显然很麻烦。

YUM 工具通过一种机制自动去解决软件包之间的依赖关系。YUM 管理工具的实现原理





是创建一个软件仓库（或称软件源），将可安装的软件包和其所有依赖包集中到仓库中（该仓库不一定是物理意义上的在一起，可以是位于任何位置，只要能被 YUM 检索到），那么，用户在使用 YUM 工具安装某软件时，YUM 就会根据 RPM 的依赖信息，自动去它的软件仓库去搜索相应的依赖包，并自动依次安装。所以，使用 YUM 安装软件，用户只需要指定安装某个软件包即可，其他过程 YUM 将自动分析完成。

YUM 默认是要去互联网上的 YUM 软件仓库上查找并下载安装软件包。如果你使用的是 Fedora 或 CentOS 这样的系统，那么直接连上互联网，你就能安装你想要的软件包了。但是 RHEL 的 YUM 软件源是需要收费的。在实验环境中，我们可以利用安装光盘来搭建一个本地源，让 YUM 自动去安装光盘中下载安装系统组件。

**示例：**搭建本地软件仓库，使用 YUM 安装软件。

**Step 01** 将 RHEL 6 的安装光盘里的内容复制到软件源目录，如 /source 目录。

**Step 02** 修改 YUM 系统配置文件，指定软件源路径。

```
[root@365linux source]# vim /etc/yum.repos.d/RHEL6local.repo
[rhel-Server]
name=Red Hat Enterprise Linux
baseurl=file:///source/Server
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

软件源共享路径支持 File、FTP、HTTP 等传输协议，其中 file 指向本地路径，而 FTP、HTTP 协议表明软件源位于网络上。在实验环境下，要在线安装软件，可以让 RHEL 使用第三方的软件源，比如 CentOS repo、RHEL repo 等。在生产环境中，稳定性要求较高的环境，推荐购买 Red Hat 的订阅服务。



#### NOTE

YUM 可以配置多个 repo 文件，每个 repo 文件中也可以有多个定义的软件源路径。但是当多个软件源 enabled 时，如果其中一个软件源访问错误，则整个 YUM 工具不能使用，会报错。



#### NOTE

当 YUM 配置使用了多个软件源时，可能在不同的软件源中存在同一个软件的不同版本，此时要安装 yum-plugin-priorities 插件，使用 priority=N 的选项指定软件源的优先级。N 的数字越大，表示优先级越低。

**REAL-WORK TIP**

使用第三方的源时，建议将配置文件中的 `enabled = 0`，而只在需要通过第三方软件源安装软件时通过 `--enablerepo=[repo]` 选项临时开启第三方的源。这样就保证了在执行 `update` 等常规操作时，第三方源不会自动开启。

**Step 03** 清除历史缓存数据（如果有）。

```
[root@365linux source] yum clean all
```

**Step 04** 列出 YUM 管理的所有软件名称和版本。

```
[root@365linux ~]# yum list
```

如果输出列表不报错，则 YUM 配置正确，就可以使用 YUM 工具安装和管理 RPM 软件包了。

示例：使用 YUM 搜索软件包。

```
[root@365linux ~]# yum search packagename
```

示例：使用 YUM 安装软件包。

```
[root@365linux ~]# yum install -y packagename
```

支持通配符。`-y` 表示安装不需要交互确认。

示例：使用 YUM 升级软件包。

```
[root@365linux ~]# yum update [packagename]
```

升级软件包。如果后面没有指定包名，则更新整个系统。

示例：使用 YUM 删除软件包。

```
[root@365linux ~]# yum remove packagename
```

示例：使用 YUM 清除 YUM 缓存数据。

```
[root@365linux ~]# yum clean all
```

清除 YUM 下载的所有缓存数据。下载的文件被放到 `/var/cache/yum` 目录下，可单独清除的项目有 `packages`、`headers`、`metadata`、`dbcache`。

示例：安装 YUM 工具的插件。

```
[root@365linux ~]# yum install yum-downloadonly yum-fastestmirror -y
```

YUM 的其他常用命令如下。

- `check-update`：检查更新。
- `upgrede`：版本升级。
- `info`：显示软件包的详细信息。
- `grouplist`：显示可使用的软件群组。
- `groupinstall`：安装一个软件群组。
- `groupupdate`：更新一个软件群组。
- `groupinfo`：显示群组的详细信息。



- `groupremove` : 卸载一个软件群组。
- `localinstall` : 使用 YUM 安装本地软件包。
- `localupdate` : 使用本地的软件包更新系统。
- `deplist` : 查询软件包的依赖关系。
- `repolist` : 激活的软件仓库的列表。

安装软件包常用参数如下。

- `--installroot=root` : 指定软件安装的路径。
- `--enablerepo=repoidglob` : 激活某个软件仓库。
- `--disablerepo=repoidglob` : 禁用某个软件仓库。
- `--exclude=package` : 排除某个软件包。
- `--disableplugin=plugin` : 禁用某个 YUM 插件。
- `--noplugins` : 不使用插件。
- `--nogpgcheck` : 不进行数字签名的检查。
- `--downloadonly` : 不更新软件包, 只下载。

YUM 的相关文件如下。

- `/etc/yum.conf` : YUM 的主配置文件, 定义了 YUM 缓存目录、日志级别与文件等。
- `/etc/yum.repos.d/` : YUM 的软件源配置文件目录。
- `/etc/yum/pluginconf.d/` : YUM 的插件配置文件目录。
- `/var/cache/yum` : YUM 的下载缓存目录。

## 11.4 搭建自己的软件仓库

如何将自己单独下载或打包的 RPM 包也能使用 YUM 安装和共享? 其实只需要将需要加入 YUM 的软件包放置到一个目录, 比如 `/source/mysoftware`, 再使用 `createrepo` 创建软件源 `repomd` (`xml-rpm-metadata`) 文件即可。命令如下:

```
[root@365linux ~]# createrepo /source/mysoftware
```

然后再配置 YUM 的配置文件中的 `baseurl` 指向该目录路径。



### 本章总结

RPM 是 Red Hat 开发并推广的一种强大的软件包管理工具, 它通过在软件包封装时加入软件的安装信息、依赖关系, 安装前后程序, 在安装时记录安装数据库, 来实现 RPM 的安装、升级、查询、检验、卸载的功能。RPM 包安装方便, 快速, 也便于管理, 在版本合适的情况下,



可优先选择。RPM 的缺点是要求安装环境的系统版本与编译打包的环境基本一致才能成功安装，且安装时灵活性和定制自由度不够。

SRPM 和 Tarball 源码包的优点在于定制性强，通过配置时参数可以定制软件功能和其他软件进行整合。由于软件安装完后文件分布在各个目录，源码包卸载起来比较麻烦，也不便于查询管理。若在安装时自定义了安装目录，则卸载时只需要将安装目录完全删除即可。

YUM 方式则是在 RHEL 5 之后的版本系列最常用的软件包管理工具。它本身是一种先进的软件包管理工具。通过在线的软件仓库，改变了原始的软件包安装方式需要手动解决依赖关系的困难。YUM 安装的软件同样可以通过 RPM 进行查询管理和卸载。

**REAL-WORK TIP**

安装、升级、卸载软件包一般使用 yum 命令，rpm 命令仅用来辅助查询和管理。。

**章节重点：**

- 了解 Linux 系统不同的软件包安装和管理方式。
- 使用 rpm 命令对 RPM 包进行安装、查询、检验、卸载。
- YUM 的配置使用和软件仓库的搭建。

**基础技能演练**

1. 使用 rpm 命令查询以下内容：

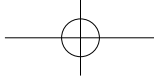
- (1) 系统中已安装的软件包和 RPM 相关的有哪些。
- (2) 系统中 ifconfig 这个命令是由哪个软件包提供的。
- (3) 确定 mount 命令在系统中的绝对路径。
- (4) 提供 mount 命令的软件包安装了哪些文件到系统里面。

2. 配置 YUM 本地软件源。

- (1) 使用 mount 命令把光盘 /dev/cdrom 挂载到 /mnt 目录。
- (2) 在 /etc/yum.repos.d 目录下创建 YUM 的配置文件 my.repo，YUM 源地址为 file:///mnt/Server。
- (3) 使用 YUM 安装软件 httpd。
- (4) 使用 YUM 的 remove 命令卸载软件包 gedit。

**提高技能实践**

1. 使用 YUM 方式为 Linux 系统安装另一套桌面环境，比如 KDE 桌面环境，安装好后，切换到 KDE 桌面，查看效果。



2. 使用网上第三方源安装一个软件包或软件包组。
  - (1) 使用 `rpmfusion` 源安装 RMVB 视频播放器。
  - (2) 安装 Firefox 的 Flash 插件。
  - (3) 安装 NTFS 文件系统的支持。

## 创建计划任务

人在工作和生活中经常制定自己的计划任务，比如计划到了某个时间点外出，或者计划好每周一的 10:00 召开会议。操作系统在运行时也需要制定各种计划任务，让它定时自动运行，比如定时关机、周期自动清理系统等。计划任务一般分为两种情况：在未来某个时间点执行的一次性计划任务；在未来某个时间点周期性循环执行的计划任务。

### 12.1 一次性计划任务

有些计划任务是只需要执行一次就完成了的。这种情况 Linux 系统使用 `atd` 这个服务来进行任务分配。使用 `at` 命令创建和管理这种类型的计划任务。

**示例：**在 2015 年 3 月 23 日执行命令 `wall "hello"`。

```
[root@365linux ~]# at 16:50 03232015
at> wall "hello"
at> <EOT>
```

任务指令输入完后按 `Ctrl+D` 组合键存盘退出。

`at` 命令指定时间的书写格式非常灵活，如：

```
at 5:30pm
at now +5minutes
at 17:30 tomorrow
```

更多 `at` 支持的时间格式，可以通过 `man at` 来查看。

**示例：**查询当前等待执行的任务。

```
[root@365linux ~]# atq
3          2010-03-23 16:50 a root
```

**示例：**删除一个等待的任务。

```
[root@365linux ~]# atrm 3
```

3 即通过 `atq` 查询得到的任务编号。

**NOTE**

默认情况下，任何用户都可以使用 at 服务，但这样对于系统不是很安全。可以通过两个文件 /etc/at.allow 和 /etc/at.deny 来进行白名单与黑名单的控制。

在 at.allow 中记录的用户才可以使用 at 服务。

在 at.deny 中记录的用户不可以使用 at 服务。

一般来说，系统只需要二者中的一个记录文件即可。

**示例：**查看系统中保存计划任务的文件。

```
[root@365linux ~]# ls /var/spool/at/
a000040142cf1a
```

at 的计划任务记录文件默认存放在 /var/spool/at 目录下。当 at 命令创建一个任务后，会在该目录下建立一个文件，该文件包含了将被执行的命令。如果将文件删除，那么对应的计划任务也会消失。

## 12.2 周期性执行的计划任务

如果计划任务需要循环来执行，那么就需要 crond 这个服务来控制。系统有很多工作是需要计划定时进行的，比如日志轮替、一些系统数据库的定期更新等，所以这个服务默认是启动的。

除了系统的计划任务，我们可以通过 crontab 命令来自定义用户的计划任务。

**示例：**root 用户创建一个计划任务，每天凌晨 3 点整执行脚本 /root/test.sh。

```
[root@365linux ~]# crontab -e
0 3 * * * /root/test.sh
```

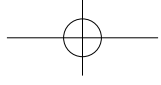
使用 crontab -e 进入计划任务的编辑界面，使用 vi 工具对任务的时间和要执行的命令进行书写或修改。5 个时间参数的含义如表 12-1 所示。

表12-1 时间参数

域	分钟	小时	日期	月份	周
范围	0 ~ 59	0 ~ 23	1 ~ 31	1 ~ 12	0 ~ 7

另外，范围取值还有特殊字符的用法。

- “\*” 表示任何时间都接受。
- “,” 表示分隔的时段都适用，如 1,3,5 表示某一个域内 1、3、5 都适用。
- “-” 表示一段时间范围都适用，如 2-5 表示 2、3、4、5 都适用。
- “/” 表示每隔多少时间，如在分钟域的 \*/5 表示每隔 5 分钟。



**示例：**列出当前用户的周期计划任务列表。

```
[root@365linux ~]# crontab -l
0 3 * * * /root/test.sh
```

**示例：**删除当前用户所有的计划任务。

```
[root@365linux ~]# crontab -r
```

如果只是要删除部分，则使用 **crontab -e** 编辑即可。

**示例：**查看系统中保存的用户周期性计划任务的文件。

```
[root@365linux cron]# ls /var/spool/cron/
root
```

cron 计划任务的记录文件存放在 /var/spool/cron 目录下。



### REAL-WORK TIP

当用户在系统中创建多个 cron 周期性计划任务时，每个任务执行的时间点应该不同并保持一定的时间间隔，比如每天执行的任务放在 02:10，则每周执行的任务放在 03:20，每月执行的任务则放在 04:10。这样避免在某些特定日子，任务集中在一起执行而消耗过多的系统资源。另外，周期性计划任务一般是安排在系统负载较低的时候自动运行，比如凌晨。

系统级需要定时运行的任务，在系统 /etc/crontab 文件中进行定义。

**示例：**查看 /etc/crontab 配置文件内容。

```
[root@365linux ~]# vim /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,
thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

默认情况下，该配置文件中并没有定义任何的周期性计划任务。用户可以添加系统级别的计划任务到该配置文件中，使用某个时间点、以某个用户的身份、执行什么任务这样的格式。

系统运行必要的计划任务事件在 /etc/cron.d 中定义和调用。





用户可以定制自己的计划任务。系统计划任务只有管理员可以编辑。

系统通过 `/etc/cron.allow` 或 `/etc/cron.deny` 来对用户使用计划任务进行白名单或黑名单的控制。



## 本章总结

计划任务是非常实用的功能，系统管理员常利用计划任务进行定期检查、数据备份、发送通知等，通过编写脚本，加入到计划任务定期自动执行。为了保证计划任务被正常执行，尤其是系统中存在必要的 `cron` 周期性系统任务，在系统中，必须要保持 `atd` 和 `crond` 这两个服务开机自动运行。

在编辑计划任务的时候，我们需要注意 3 个问题：

- (1) 避免不必要的屏幕输出信息发送。
- (2) 多个任务要时间交错执行。
- (3) 命令使用绝对路径。



### REAL-WORK TIP

在系统中计划任务是定时自动运行的，所以计划任务也是系统被入侵后自动执行恶意脚本的手段之一。

章节重点：

- `at` 的使用。
- `cron` 的使用。



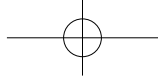
## 基础技能演练

1. 创建 `at` 计划任务。

- (1) 计划本月底发一封邮件给用户 `user001`（提示：发邮件使用 `mail` 命令）。
- (2) 5 小时以后关闭系统。

2. 创建 `cron` 计划任务。

- (1) 每天 18:30 定时关机。
- (2) 每周六的凌晨 4 点，检查系统内是否有 `SUID` 或 `SGID` 的文件，将结果记录到 `/root/sfile.txt`。



## 提高技能实践

如果一个 `at` 计划任务在它将被触发执行的时间点，系统是关机的，即系统错过了那个时间点，那么这个 `at` 计划任务在下次系统开机后会被执行吗？`cron` 计划任务发生同样的情况呢？



## 第 13 章

# 监控系统资源和运行状态

系统管理员要定期对运行中的系统进行健康检查，通过系统进程的状态、进程对系统资源的占用，分析系统的运行状况，解决故障或排除潜在问题。本章将介绍一些常用的系统检测命令。

## 13.1 查看和管理系统进程

### 1. 进程的概念

进程是 Linux 用来表示正在运行的程序的一种抽象的概念。Linux 系统上所有运行的东西都可以称之为进程，包括用户任务、批处理进程和守护进程。进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。

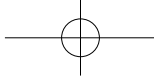
进程一般来说由程序产生，但不等同于程序。进程是一个动态的概念。一个程序可以启动多个进程。

运行一个程序则触发一个事件，这个事件被称为作业。在 Bash 的环境下，可以对作业进行管理，比如将作业放置到后台或前台，暂停或程序启动。

系统最开始的进程是 init。init 的 PID（进程号）始终为 1。

一个进程可以产生另外一个进程。如在登录的 Bash 下执行 cat 命令，则 cat 就是 Bash 的子进程。通过 pstree 或 ps -ely（比较 PID 与 PPID）可以清楚地知道进程之间的关系。除了 init 这个进程外，所有的进程都有父进程。当一个进程不正常时，可以通过追踪它的父进程来确定进程的作用及采取适当的措施。如果父进程在子进程终止之前消亡，那么这个“孤儿”将会被 init “收养”。

一般来说，子进程会继承父进程的权限。进程的 UID 就是其创建者的 UID。那么，系统只允许进程所有者和 root 对其进行操作。同时 EUID 规定了进程对哪些资源和文件有访问权限。



示例：查看进程状态。

```
[root@365linux ~]# ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	2064	624	?	Ss	Mar03	0:01	init [3]
root	2	0.0	0.0	0	0	?	S<	Mar03	0:00	[migration/0]
root	3	0.0	0.0	0	0	?	SN	Mar03	0:00	[ksoftirqd/0]
root	4	0.0	0.0	0	0	?	S<	Mar03	0:00	[watchdog/0]
root	5	0.0	0.0	0	0	?	S<	Mar03	0:00	[events/0]

ps 用来查看进程信息，功能强大，用法也比较复杂，最常用的参数有 ps aux、ps -ef、ps -ely。

选项说明如下。

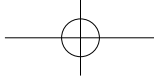
- a：显示所有进程，但不包括不属于任何终端的进程。
- u：显示进程拥有者、状态、资源占用等的详细信息（注意有“-”和无“-”的区别）。
- x：显示没有控制终端的进程。
- -e：显示所有进程。
- -f：显示全格式。
- -l：显示长格式。
- -y：与 -l 配合使用。

输出中每列的意义如下。

- USER：进程拥有者，即创建者。
- PID：进程 ID。
- %CUP：占用 CPU 的百分比。
- %MEM：占用内存的百分比。
- VSZ：占用虚拟内存的大小（KB）。
- RSS：占用物理内存的大小（KB）。
- TTY：当前进程执行终端号。
- STAT：该进程的状态。D 表示不可中断的睡眠状态，除非发生指定事件，否则不会被唤醒；R 表示正在执行中；S 表示睡眠状态，可被某些信号唤醒；T 表示暂停执行；Z 表示僵尸进程，例如已终止但未能被父进程回收的子进程；N 表示低优先序进程；< 表示高优先级进程；L 由内存分页分配并锁在内存中；s 为一个会话的前导字符；l 为多线程；+ 为前台进程。
- START：进程开始时间。
- TIME：进程实际使用 CPU 运行的时间。
- COMMAND：进程实际命令。

查看进程的其他常用命令如下。

- pstree 是以进程树的形式显示进程之间的父子关系。



- **pgrep** 是以进程名字或属性（如进程 UID）来显示查找进程的 PID。

示例：查看系统即时运行状态（资源管理器）。

```
[root@365linux ~]# top
top - 01:44:09 up 5:17, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 65 total, 1 running, 64 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 515444k total, 435564k used, 79880k free, 58088k buffers
Swap: 1052248k total, 0k used, 1052248k free, 327472k cached

  PID  USER  PR   NI  VIRT  RES  SHR  S   %CPU  %MEM    TIME+  COMMAND
    1   root   15    0   2064   624   536  S    0.0   0.1    0:01.19  init
    2   root   RT   -5    0     0     0  S    0.0   0.0    0:00.00  migration/0
    3   root   34   19    0     0     0  S    0.0   0.0    0:00.00  ksoftirqd/0
    4   root   RT   -5    0     0     0  S    0.0   0.0    0:00.00  watchdog/0
```

**top** 命令是 Linux 下常用的性能分析工具，能够实时地显示系统资源占用状况。

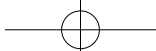
**top** 命令显示结果上半部分是系统整体的统计信息，分别表示：

- 当前时间是 01:44:09，系统运行时间 5 小时 17 分，当前有 3 个用户登录，系统负载：1 分钟 0.00，5 分钟 0.00，15 分钟 0.00。
- 进程总数 65 个，1 个在运行，64 个在睡觉，0 个暂停，0 个僵死。
- CPU 占用百分比，用户空间占用 0.0%，内核空间占用 0.0%，用户进程空间内改变过优先级的进程占用 CPU 的百分比为 0.0%，CPU 空闲 99.7%，等得输入 / 输出的 CPU 时间为 0.0%。
- 物理内存大小是 512 444KB，435 564KB 被使用了，79 880KB 空闲，58 088KB 用作缓存。
- 交换空间的大小是 1 052 248KB，0KB 被使用了，1 052 248KB 空闲，缓冲交换区的大小是 327 472KB。

**top** 命令显示结果的下半部分是各进程的详细信息，每列的含义如表 13-1 所示。

表13-1 top进程信息

序号	列 名	含 义
A	PID	进程 PID
E	USER	进程拥有者的用户名
H	PR	优先级
I	NI	NICE 值，负值表示高优先级，正值表示低优先级
O	VIRT	进程使用的虚拟内存总量，单位为 KB。VIRT=SWAP+RES
Q	RES	进程使用的、未被换出的物理内存大小，单位为 KB。RES=CODE+DATA
T	SHR	共享内存的大小，单位为 KB
W	S	进程状态
K	%CPU	上次更新到现在 CPU 时间使用百分比
N	%MEM	进程使用内存百分比



续表

序号	列 名	含 义
M	TIME+	进程使用 CPU 时间
X	COMMAND	命令名 / 行

这是 `top` 命令默认显示的列。可以在 `top` 输出界面下，按 `f`（或 `F`）键显示所有的列，按相应的序列号显示或隐藏对应的列，按 `Enter` 键确认。

在 `top` 的执行过程中，还可以使用以下的按键命令。

- `?`：显示在 `top` 中可以输入的按键命令。
- `P`：按 CPU 的使用资源排序显示。
- `M`：按内存的使用资源排序显示。
- `N`：按 PID 来排序。
- `T`：按该进程使用的 CPU 时间累积排序。
- `k`：给某个 PID 一个信号（Signal），默认值是信号 15。
- `r`：重新安排一个进程的优先级别。
- `i`：忽略（显示）闲置和僵死的进程。
- `S`：切换到累计模式。
- `s`：改变两次刷新之间的时间。默认是 5 秒。
- `o`：按 `o` 键可以改变列的显示顺序。按小写的 `a ~ z` 键可以将相应的列向右移动，而按大写的 `A ~ Z` 键可以将相应的列向左移动。最后按 `Enter` 键确定。
- `O`：按 `O` 键后再按 `a ~ z` 键可以将进程按照相应的列进行排序。按大写的 `R` 键可以将当前的排序倒转。
- `l`：切换显示平均负载和启动时间信息。
- `m`：切换显示内存信息。
- `t`：切换显示进程和 CPU 状态信息。
- `c`：切换显示命令名称和完整命令行。
- `W`：将当前设置写入 `~/.toprc` 文件中。这是写 `top` 配置文件的推荐方法。
- `q`：退出程序。

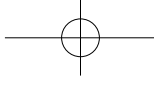
`top` 命令常用的选项如下。

- `-d`：后面可以接秒数，指定每两次屏幕信息刷新之间的时间间隔。
- `-p`：指定某个进程来进行监控。
- `-b -n`：以批处理方式执行 `top` 命令。通常使用数据流重定向，将处理结果输出为文件。

## 2. 进程控制

示例：以 `nice` 值为 `-20` 运行 `top` 命令。

```
[root@365linux ~]# nice -n -20 top
```



Linux 系统中每个进程都有一个优先级 (PR)，PR 值越小，表示优先级越高，则越被优先执行。nice 值是我们设定的值，取值范围 root 为 -20~19，普通用户为 0~19。nice 值的正负影响到 PR 值，所以我们可以设定 nice 值来提升或降低进程的优先级。普通用户仅能降低（增加 nice 值）属于自己进程的优先级。

示例：临时修改进程的优先级。

```
[root@365linux ~]# renice 10 3942
```

将 PID 为 3942 的进程的 nice 值改为 10。前文提到，也可以通过 top 命令中的 r 按键来修改 nice 值。

示例：终止进程。

```
[root@365linux ~]# kill 3942
```

kill 命令是发送一个信号给进程。默认是发送 15 (TERM) 终止。

示例：强制中断 PID 为 3942 的进程。

```
[root@365linux ~]# kill -9 3942
```

示例：平滑重启进程。

```
[root@365linux ~]# kill -HUP 3942
```

示例：终止所有 httpd 启动的进程。

```
[root@365linux ~]# killall httpd
```

示例：结束用户 user001 的所有进程。

```
[root@365linux ~]# pkill -9 -U user001
```

示例：查看哪个进程在使用该文件。

```
[root@365linux ~]# fuser /var/run/atd.pid
```

示例：查看哪些进程监听了 80 端口。

```
[root@365linux ~]# lsof -i :80
```

示例：查看 httpd 相关的进程 PID。

```
[root@365linux ~]# pidof httpd
```

## 13.2 收集系统运行状态信息

示例：报告设备或分区的 I/O 相关的统计，每 3 秒钟刷新一次，报告两次。

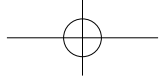
```
[root@365linux ~]# iostat -d 3 2
```

示例：报告处理器相关的统计。

```
[root@365linux ~]# mpstat
```

示例：报告虚拟内存相关的统计。

```
[root@365linux ~]# vmstat
```



示例：报告系统运行时间和系统负载。

```
[root@365linux ~]# uptime
```

示例：收集、报告或者保存系统活动信息。

```
[root@365linux ~]# sar 2 3
```

示例：以多种格式显示 sar 收集的信息。

```
[root@365linux ~]# sadf
```

## 13.3 虚拟文件系统

proc 目录是由 procfs 的文件系统所产生出来的，是 Kernel 加载后，在内存里面建立的一个虚拟目录，包含系统运行时内核信息、进程信息、硬件信息、网络设置、内存使用等，实际上就是系统运行时的内核状态在用户空间的显示。为了保障系统的稳定性，proc 目录内的文件不能进行写操作，但 /proc/sys 目录下的一些文件可以进行重定向开关和更改，以达到在系统运行时调整和优化内核参数的目的。

示例：查看 proc 文件系统的挂载状态。

```
[root@365linux ~]# mount
/dev/sda3 on / type ext3 (rw)
proc on /proc type proc (rw)
...
```

示例：将 proc 挂载到另外的目录下。

```
[root@365linux vm]# mount -t proc /mnt /mnt
```

proc 目录内容介绍如下。

- number：代表目前正在系统中运行的程序。
- cpuinfo：CPU 硬件信息。
- cmdline：加载 Kernel 执行的相关参数。
- devices：/dev 目录中设备文件分类方式。
- filesystems：系统已加载的文件系统。
- interrupts：系统上 IRQ 分配状态。
- ioports：系统上设备对应的 I/O 地址。
- kcore：系统上的物理内存大小。不能 cat。
- meminfo：内存信息。
- modules：系统上使用的模块。
- partitions：系统分区信息。
- net：网络相关的文件。





- `scsi` : SCSI 设备信息。
- `sys` : 核心配置参数。

`sys` 里面文件的设置值是在系统核心里面使用的, 部分参数可以实时更改。

**示例** : 实时打开内核网络转发的功能。

```
[root@365linux ipv4]# echo "1" >/proc/sys/net/ipv4/ip_forward
```

通过 `proc` 调整内核参数是立即生效, 但不会保存 (即下次重启失效)。用户可以通过修改 `/etc/sysctl.conf` 文件设置内核参数默认值。



## 本章总结

在本章学习过程中, 我们要学会对系统信息进行收集、分析, 了解系统运行状况。

对进程的关注是 Linux 系统管理最关键的部分, 查看每个进程的运行状况, 以及对系统资源的占用, 对优化系统性能、防范系统潜在的安全威胁都是非常重要的。

要熟悉系统 `/proc` 虚拟文件系统的内容, 特别对 `/proc/sys` 目录有所留意。通过修改 `/proc/sys` 内文件的设置值可以直接修改内核运行参数, 实时地改变系统运行状态。也可以通过对 `/etc/sysctl.conf` 文件的修改来更改系统内核运行参数的默认值达到按需定制系统, 提高系统性能和安全性的要求。

**章节重点 :**

- 系统进程的控制。
- 系统状态检测。
- 系统内核信息查看与修改。



## 基础技能演练

1. 如何踢掉一个从远程登录的用户?
2. 如果一个前台进程僵死占据了终端, 应该如何结束它?
3. 打开当前系统的 IPv4 转发功能, 并使之永久生效。



## 提高技能实践

完成一次完整的服务健康巡检工作, 巡检内容和巡检报告范本另行提供。



## 第14章

# 系统日志分析

日志是系统守护进程、内核及各种工具在运行的过程中产生的信息记录，目的是为了帮助管理员能够通过读取系统日志来获得系统信息、检查错误原因、追踪网络威胁等。

## 14.1 查看系统日志

在 Linux 系统中，日志记录文件默认存放在系统 `/var/log` 目录下，大部分可以通过文本查看命令直接查看。但出于安全考虑，与连接登录有关的日志以二进制文件形式存在，需要用相关命令来查看。常用的日志文件如下。

- `anaconda*.log`：系统安装日志。
- `audit`：存放系统安全审计的日志，通常与 SELinux 有关。
- `boot.log`：系统引导过程中日志；从系统初始化开始的日志，不包含内核引导的日志。且只保留最近一次的系统引导日志记录。
- `btmp`：用户登录系统失败的日志记录，要使用 `lastb` 命令进行输出查看。
- `consoleKit/history`：图形界面中普通用户管理员授权的历史记录。
- `cron`：周期性计划任务日志。
- `cups`：打印设备使用日志。
- `dmesg`：引导时内核加载的日志。
- `dracut.log`：在安装系统时创建 initial ramdisk images 产生的日志。
- `gdm`：图形界面相关的日志。
- `lastlog`：显示系统中所有用户最后登录的时间，只能使用 `lastlog` 命令来查看。
- `libvirt`：虚拟化相关的日志。
- `miallog`：邮件服务器日志。



- `messages` : 系统通用日志记录文件（没有特别指定日志文件的日志信息都会记录到此文件）。
- `ntpstats` : 时间同步服务相关的日志。
- `pm-powersave.log` `pm-suspend.log` : 电源管理（挂起和休眠）相关的日志。
- `prelink/prelink.log` : 记录库的分配的虚拟地址空间位置。
- `rhsm` : Red Hat 授权认证系统管理相关的日志。
- `sa` : 收集系统运行的状态信息。
- `secure` : 涉及账号或密码的安全相关的信息。
- `wtmp` : 系统账号登录成功的日志记录，必须要用 `last` 命令查看。
- `Xorg*.log` : Xorg 图形服务相关的日志。
- `yum.log` : YUM 安装软件的日志。

很多服务日志会记录在 `/var/log/service_name` 目录下归类管理，日志的存放位置可以自定义。

## 14.2 日志产生机制

`syslog` 是一个日志记录系统，在 RHEL 6 中它升级为 `rsyslog`，是 `syslog` 的多线程版本。它由一个守护程序组成，随系统启动并持续运行。它能接受系统应用产生的日志信息，并且根据 `/etc/rsyslog.conf` 的配置信息分类处理，将不同类别、不同程度的消息发送到不同的地方。

`/etc/rsyslog.conf` 的基本格式为：选择域 <tab> 动作域。

选择域指明发送日志的“设备”和消息的严重性级别；动作域指明如何处理这一消息（记录到文件或发送到设备）。

**示例：**来自邮件系统的 `info` 级别以上（包括 `info`）的消息记录到 `/var/log/maillog` 文件中。

```
mail.info /var/log/maillog
```

设备的名称和消息严重性必须从已定义的值中选择。设备分别为内核、常用应用程序组及本地编写的程序进行定义。任何其他程序则归为普通设备“用户（`user`）”一类。

设备名称可以是 `*` 和 `none`，分别表示“所有（除了 `mark`）”和“没有”。多个设备用“`,`”隔开，多个选择域使用同一动作域可以“`;`”连接。

**示例：**所有设备 `info` 级别以上（包括 `info`）的消息都记录到 `messages` 文件，但除了 `mail`、`authpriv`、`cron`。

```
*.info;mail.none;authpriv.none;cron.none /var/log/messages
```

`rsyslog` 识别的发送消息的设备如表 14-1 所示。



表14-1 发送消息的设备

设 备	含 义
auth	与安全 and 授权有关的命令
authpriv	敏感 / 保密的授权消息
cron	与计划任务有关
daemon	与系统守护进程有关的命令
ftp	FTP 相关
kern	与 Kernel 相关
local 0 ~ 7	本地消息的 8 种类型
lpr	打印机有关
mail	与邮件系统有关
mark	由 syslogd 定期产生的时间戳
news	Usenet 新闻系统
syslog	syslog 本身产生的消息
user	用户进程（如果没有指定，这将是默认值）
uucp	为 UUCP 保留，并未使用

rsyslog 的严重性级别定义如表 14-2 所示。

表14-2 日志严重性级别

严重性级别	含 义
debug	仅供调试
info	提供基本的消息
notice	需要注意的消息
warning (warn)	警告消息
error(err)	错误消息
crit	临界状态
alert	严重紧急状态
emerg (panic)	恐慌崩溃状态

使用 \* 表示对某个设备的所有消息进行处理，使用 none 表示忽略某个设备的消息。

设备和严重性级别之间用 “.” 进行连接，表示严重性要大于等于定义的级别才会被处理。

Linux 支持 “=” 和 “!”，分别表示只匹配定义级别，除定义的之外的级别。

rsyslog 支持的对消息的处理动作如表 14-3 所示。

表14-3 消息的处理动作

处理动作	含 义
filename	把消息记录到本机上的一个文件里
device	把消息输出到设备，如打印机
@hostname	通过主机名把消息发给远程上的主机
@ipaddress	通过 IP 地址把消息发给远程上的主机

续表

处理动作	含 义
fifoname	把消息通过管道发送
username	把消息发送给在线用户
*	把消息发送给在线的所有用户

Linux 中的许多服务和应用程序都使用 syslog，常见的有：cron、cups、ftpd、inetd、imapd、login、lpd、named、ntpd、passwd、sendmail、ssh、su、sudo、syslog、tcpd、vmlinux、xinetd。

### 14.3 日志切割与轮替

由于日志消息总是不断追加，日志文件变得越来越大，不仅占用了有效的磁盘空间，而且对于庞大的文本文件的写入也会极大地降低系统的性能。如果一次性将旧的日志删除，那会给系统管理员想查看以往的日志造成极大的不便。所以我们需要一个规则，定期地将日志切割轮替，保留一段时间内的日志信息，并将更旧的日志删除。logrotate 就负责这一工作。

logrotate 工具配合计划任务进行周期性的日志切割。

**示例：**关于 logrotate 的系统计划任务。

```
[root@365linux ~]# cat /etc/cron.daily/logrotate
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

**示例：**手动触发日志轮替，并显示轮替的过程信息。

```
[root@365linux log]# logrotate -vf /etc/logrotate.conf

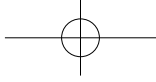
/etc/logrotate.conf 配置文件定义了轮替的规则。
```

**示例：**查看 logrotate.conf 配置文件信息。

```
[root@365linux ~]# grep -Ev '^#' /etc/logrotate.conf
```

主要参数详解如下。

weekly	# 默认每周对日志文件进行一次轮替
rotate 4	# 保留 4 个日志文件
create	# 创建新的日志文件
include /etc/logrotate.d	# 包括该目录下的文件
/var/log/wtmp {	# 对 wtmp 日志文件的单独定义



```
monthly                # 每个月进行一次轮替
minsize 1M             # 或者是日志大小超过 1MB 时进行轮替
create 0664 root utmp  # 新建日志文件的权限和用户及所属组
rotate 1               # 只保留一个日志文件
}
```

**示例：**查看子配置文件目录下的 `syslog` 单独定义的轮替规则。

```
[root@365linux logrotate.d] # cat /etc/logrotate.d/syslog
/var/log/messages /var/log/secure /var/log/maillog /var/log/spooler /var/
log/boot.log /var/log/cron {      # 定义了多个日志文件的轮替
    sharedscripts                # 与 endscript 搭配，可以定义轮替前或轮替后执行的命令
    postrotate                  # 以下是轮替后执行的命令。如要在轮替前执行，则使用 prerotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null
|| true
        /bin/kill -HUP `cat /var/run/rsyslogd.pid 2> /dev/null` 2> /dev/null
|| true
    endscript
}
```



## 本章总结

一般来说，系统进程的运作、系统故障和安全攻击都会留下日志信息，管理员可以通过查看和分析日志来找出系统异常原因，防范恶意攻击。在查看和分析日志时，可以通过时间戳截断需要的时间段的日志进行分析，或者采用关键字过滤提高效率。当需要分析的日志量非常大，且需要统计日志数据，如对 **Web** 服务器的访问日志进行分析时，则需要借助专门的日志分析工具，通过图表直观地展示。

对于管理较多的 **Linux** 服务器来说，可以单独提供一台中央日志记录主机，将其他服务器产生的较重要的消息通过远程日志功能统一集中到中央日志记录主机上管理。这样做一方面可以方便系统管理员操作，另一方面可以在一定程度上防止服务器被恶意入侵后，侵入者删除本地日志痕迹，无迹可循。

### 章节重点：

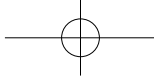
- 查看和分析系统常用的日志文件。
- 日志产生的机制。
- 日志切割与轮替机制。



## 基础技能演练

### 1. 查看系统日志。

(1) 查看系统用户登录成功和登录失败的日志。



(2) 查看系统近期安装软件的日志。

(3) 查看系统启动的日志。

## 2. 日志切割轮替。

(1) httpd 服务访问日志文件 `/var/log/httpd/access_log`，我们要自定义它的轮替规则，要求日志文件一个月进行一次轮替，或者文件大小超过 10MB 就进行轮替，保留 3 个日志记录文件。

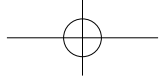
(2) 另外，假设该日志文件内容涉及安全信息，我们为增强安全性，给该文件加上 `+a` 的隐藏属性，使其只能追加数据，而不能删除修改。

(3) 如何定义日志轮替规则，以满足上述要求？



## 提高技能实践

开启普通用户的 `sudo` 功能，实现 `sudo` 的自定义日志（详细记录授权用户使用 `sudo` 执行了什么命令，并单独记录到 `/var/log/exsudo.log` 文件中），并每月进行轮替，保留 3 个备份，且单独将 `sudo` 的日志发送到另外一台远程日志主机上作为备份。



## 附录A

# VMware Workstation使用入门

本附录主要介绍如何安装和使用 VMware Workstation，创建一个典型虚拟机，以及执行虚拟机操作的常用功能。

## A.1 什么是 VMware Workstation

VMware Workstation 是适用于台式机和笔记本电脑的最先进虚拟化软件，可大幅提高专业技术人员及其所在组织的生产效率。VMware Workstation 的工作原理是：创建完全隔离且安全的虚拟机来封装操作系统及其应用。VMware 虚拟化层将物理硬件资源映射到虚拟机的资源，所以每一台虚拟机都有自己的 CPU、内存、磁盘和 I/O 设备并且完全等同于一台标准的 x86 计算机。VMware Workstation 安装在主机操作系统上，并通过继承主机的设备支持来提供广泛的硬件支持。

能够在标准 PC 上运行的任何应用都可以在 VMware Workstation 上的虚拟机中运行。VMware Workstation 相当于一台完整的 PC，拥有完整的网络连接和设备。

主机操作系统环境将 VMware Workstation 视为应用。在虚拟机上安装时不需要对客户操作系统进行任何修改。应用在客户操作系统中的运行方式与它们在主机的运行方式完全相同。

使用 VMware Workstation 可以在主机操作系统上的虚拟机中运行另一个操作系统。无须进行双启动，就可同时运行两个操作系统，而且只需单击一下即可从一个操作系统无缝切换到另一个操作系统。

## A.2 软硬件需求

VMware Workstation 可在使用 Intel 和 AMD 处理器的基于 x86 的标准硬件及大多数





Windows 或 Linux 主机操作系统上运行。VMware 建议以下配置。

### A.2.1 系统要求

- 64 位 x86 Intel Core™ 单核处理器或同等级别的处理器，AMD Athlon™ 64 FX 双核处理器或同等级别的处理器。
- 1.3 GHz 或更快的核心速度。
- 至少 2 GB RAM，建议 4 GB。

Workstation 安装：应用需要 1.2 GB 磁盘空间。每台虚拟机需要额外的硬盘空间。有关具体客户操作系统的建议磁盘空间，请咨询供应商。

### A.2.2 主机操作系统\*（32位和64位）

- Windows 8。
- Windows 7。
- Windows Vista。
- Windows XP SP3。
- Windows Server 2012。
- Windows Server 2008。
- Windows Server 2003 Standard。
- Windows XP Home Edition with SP2。
- Ubuntu 8.04 及更高。
- Red Hat Enterprise Linux 5 及更高。
- CentOS 5.0 及更高。
- Oracle Linux 5.0 及更高。
- openSUSE 10.2 及更高。
- SUSE Linux 10 及更高。

\* 有关受支持的主机和客户操作系统的最新列表，请访问《VMware 兼容性指南》。

### A.2.3 对于虚拟机中的 Windows 7 Aero 图形支持

- 3 GB RAM（主机 PC）。
- Intel 双核，2.2 GHz 及更高或 AMD Athlon 4200+ 及更高。
- NVIDIA GeForce 9900GT 及更高或 ATI Radeon HD 2600 及更高。

**REAL-WORK TIP**

如果您需要在虚拟机中安装运行一个 64 位的客户机操作系统（Guest Operating System，即在虚拟机中运行的操作系统），需要以下 CPU 的支持：

- AMD CPU 支持 long mode segment-limit。
- Intel CPU 支持 VT-x（在主机 BIOS 中开启）。

主机操作系统（Host Operating System，真实物理机操作系统）必须是 64 位。如果客户机也需要开启虚拟化功能（即虚拟化嵌套），则需要在 VMware Workstation 的虚拟硬件设置中设置处理器开启 Intel VT-x 或 AMD-V 功能。

## A.3 安装和使用VMware Workstation

### A.3.1 获得VMware Workstation软件和许可证号

请到 VMware 官方网站下载相应的软件，目前最新版本产品信息如下：

```
VMware Workstation for Windows
Name: VMware-workstation-full-10.0.1-1379776.exe
文件大小：491M
文件类型：exe
发行日期：2013-10-24
内部版本号：1379776
Product installation including VMware Tools for all operating systems.
Windows 32-bit and 64-bit
MD5SUM: ccdb65d8e24137e2407d4c6bc6belb01
SHA1SUM: 7c2c58d6f214073933be2fe93a5ccd308f969fb8

VMware Workstation 10.0.1 for Linux
...
```

**NOTE**

下载完成后首先要校验下载文件的 MD5SUM 或 SHA1SUM 的值是否匹配官方值，以保证下载文件的完整性和安全性。

VMware Workstation 是一个商业软件，在使用时必须为每一个安装提供一个 License Key（许可证）。如果您在安装的时候没有输入 Workstation License Key，可以在安装完成后，选择菜单“帮助”→“输入许可证密钥”，如图 A-1 所示，在弹出的对话框中输入正确的许可证号。



图A-1 输入许可证密钥

**NOTE**

如果您没有输入可用的许可证号，Workstation 会在您每次尝试启动一个虚拟机系统时提示您（会阻止虚拟机启动）。

**REAL-WORK TIP**

在大多数测试场景中，可以使用免费的 VMware Player（Workstation 的简化版）来替代 Workstation，或者使用免费开源软件 VirtualBox（另一个著名的虚拟机软件）。如果您的工作需要必须使用 VMware Workstation，请联系 VMware 公司购买。

## A.3.2 安装VMware Workstation

### 1. 在 Windows 主机中安装 Workstation

步骤如下：

**Step 01** 以管理员用户或本地管理员组成员用户登录 Windows 主机系统。

**Step 02** 双击下载的安装文件，例如 VMware-workstation-xxxx-xxxx.exe（xxxx-xxxx 是版本和编译号）。

**Step 03** 选择安装选项：Typical 或 Custom。

**Step 04** 根据安装向导完成安装。

### 2. 在 Linux 主机中安装 Workstation

安装过程略。

**REAL-WORK TIP**

在安装完成后，可以通过选择菜单“编辑”→“首选项”，在弹出的对话框中对 Workstation 的属性进行设置，比如设置“虚拟机的默认位置”。



### A.3.3 升级VMware Workstation

您可以从 Workstation 4、5.x、6.x 或 7.x 升级到当前的版本，通过直接运行新版本的安装程序覆盖安装，安装程序会先自动卸载掉之前的安装版本。要使用最新的功能，之前由旧版本创建的虚拟机必须升级到当前的版本。



#### REAL-WORK TIP

在跨版本升级的过程中，Workstation 的功能可能存在兼容性问题，比如快照、虚拟硬件等。升级前须做好虚拟机的备份，并参照官方文档升级向导步骤进行。

### A.3.4 卸载VMware Workstation

在 Windows 中卸载 Workstation，可直接运行安装程序，在“程序维护”界面中选择“移除”，然后根据向导完成卸载；或者在 Windows 系统控制面板“卸载或更改程序”界面，选择 VMware Workstation 并执行“卸载”命令。

### A.3.5 启动VMware Workstation

在 Windows 系统中，双击桌面上 Workstation 的快捷方式图标，或者单击“开始”菜单 / 快速启动栏图标启动 VMware Workstation 软件。启动后界面如图 A-2 所示。



图A-2 VMware Workstation初始界面



## A.4 创建虚拟机

单击主页选项卡中的“创建新的虚拟机”图标,或者单击“文件”→“创建新的虚拟机”,弹出“新建虚拟机向导”,选择“典型”则可以通过几个简单的步骤创建 Workstation 10.0 虚拟机;选择“自定义”手动创建更多高级选项的虚拟机。

这里以“自定义”方式创建一台新的为 Linux 系统准备的虚拟机,介绍各个选项的选择,如图 A-3 所示。



图A-3 新建虚拟机向导

### NOTE

此外您还可以从一台存在的虚拟机或虚拟机模板克隆一台新的虚拟机;导入开放虚拟化格式(OVF)和第三方的虚拟机;创建远程虚拟机。

步骤如下:

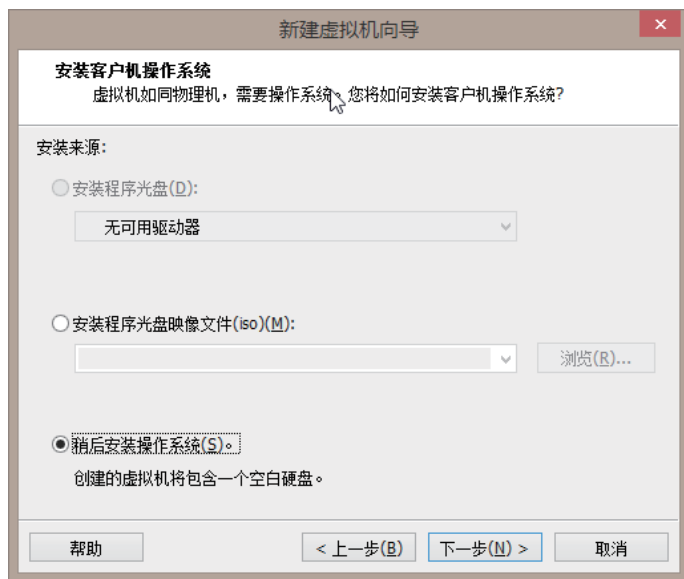
**Step 01** 选择虚拟机硬件兼容性,如图 A-4 所示。



图A-4 选择虚拟机硬件兼容性

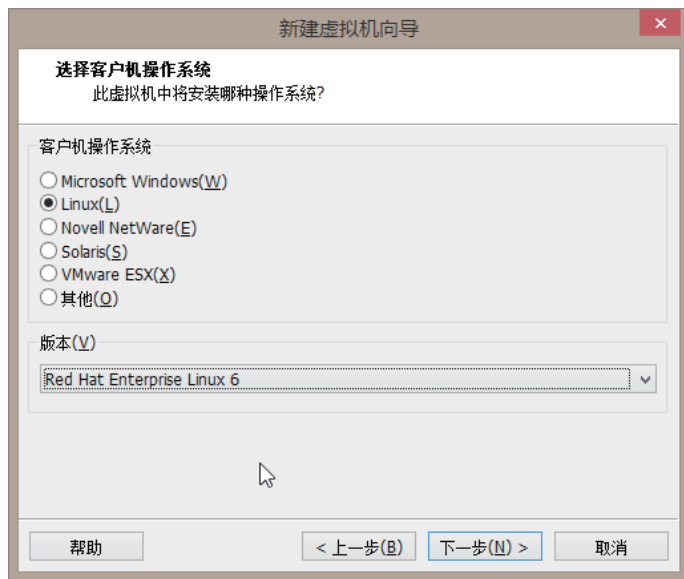


**Step 02** 安装客户机操作系统，如图 A-5 所示。



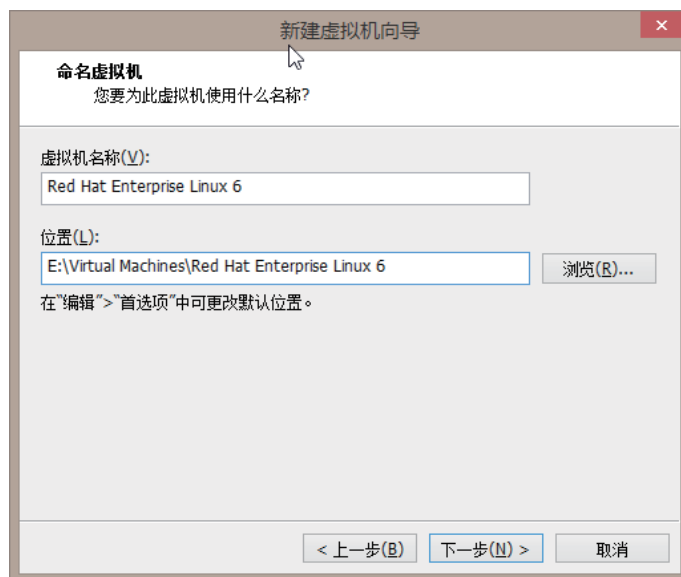
图A-5 安装客户机操作系统

**Step 03** 选择客户机操作系统，如图 A-6 所示。



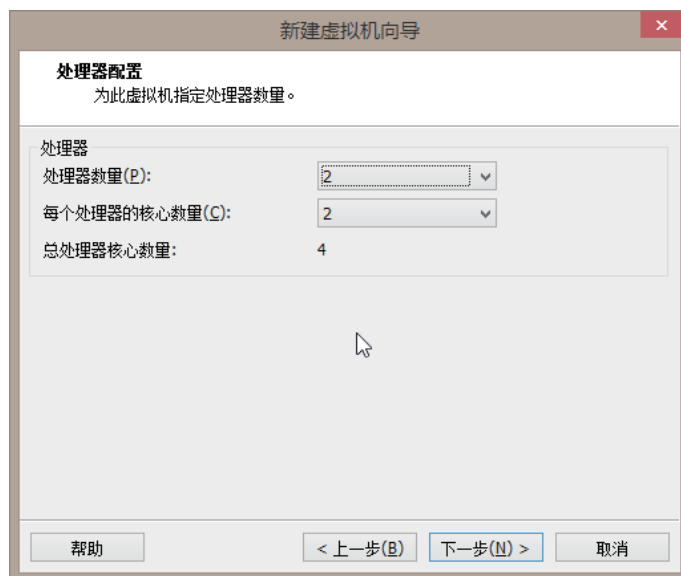
图A-6 选择客户机操作系统

**Step 04** 命名虚拟机，如图 A-7 所示。



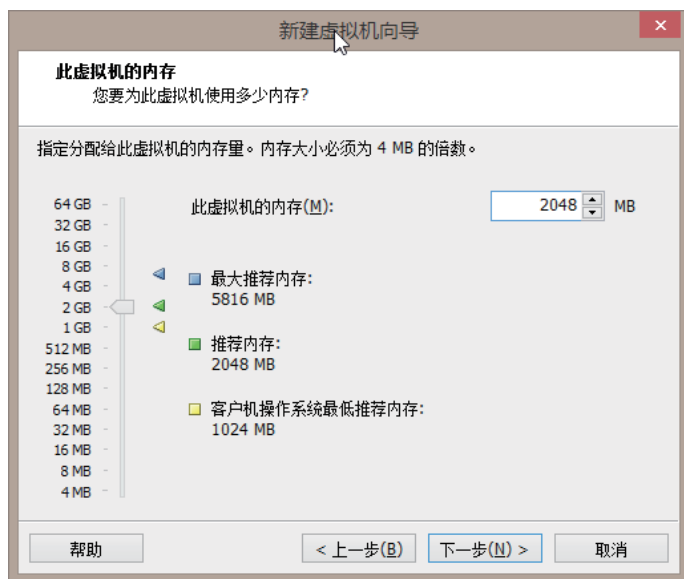
图A-7 命名虚拟机

**Step 05** 处理器配置，如图 A-8 所示。



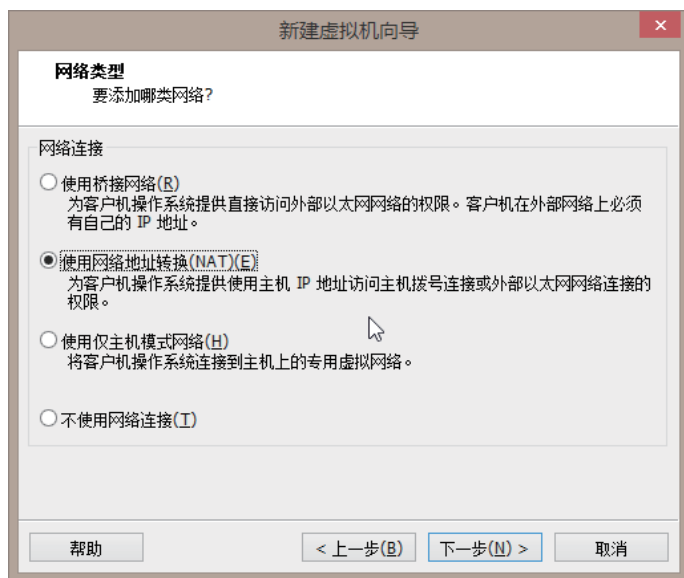
图A-8 处理器配置

**Step 06** 内存的配置，如图 A-9 所示。



图A-9 此虚拟机的内存

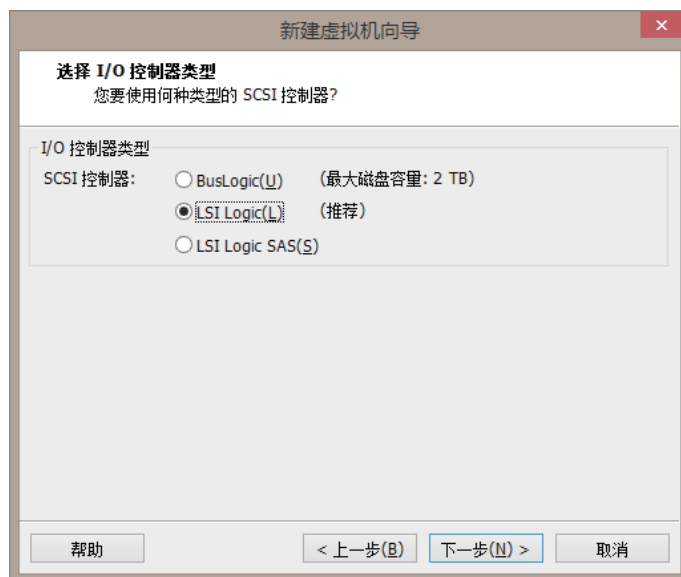
**Step 07** 网络类型（关于网络连接类型请参见 A.5.4 节），如图 A-10 所示。



图A-10 网络类型

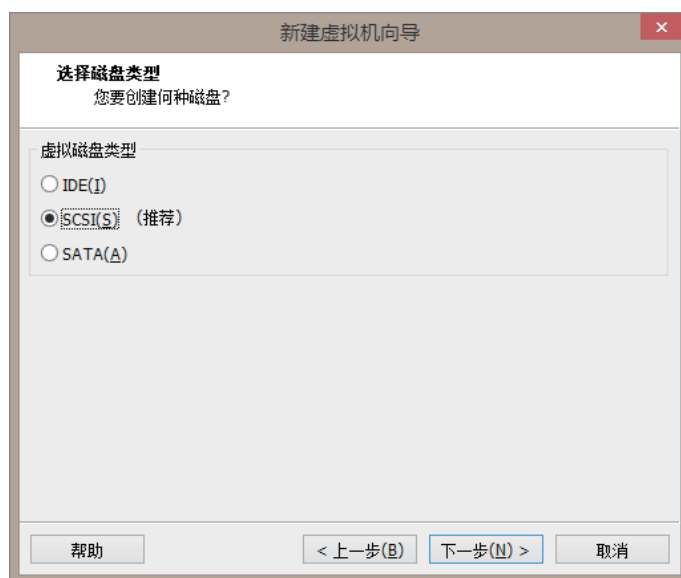
**Step 08** 选择 I/O 控制器类型，如图 A-11 所示。





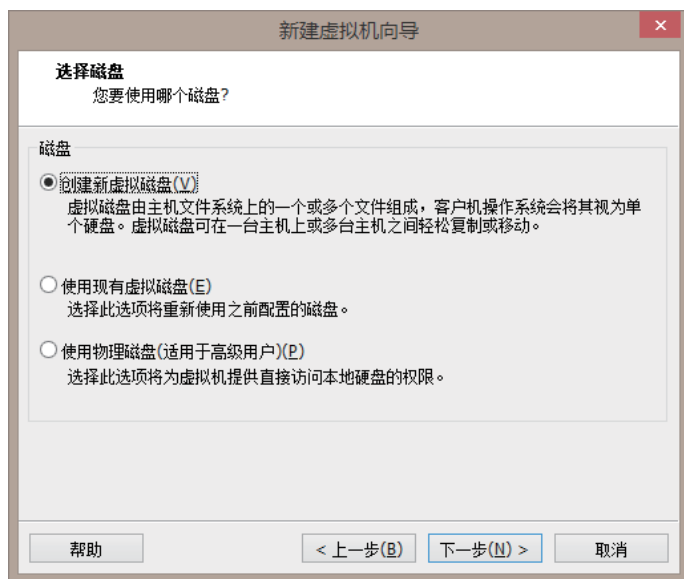
图A-11 选择I/O控制器类型

**Step 09** 选择磁盘类型，如图 A-12 所示。



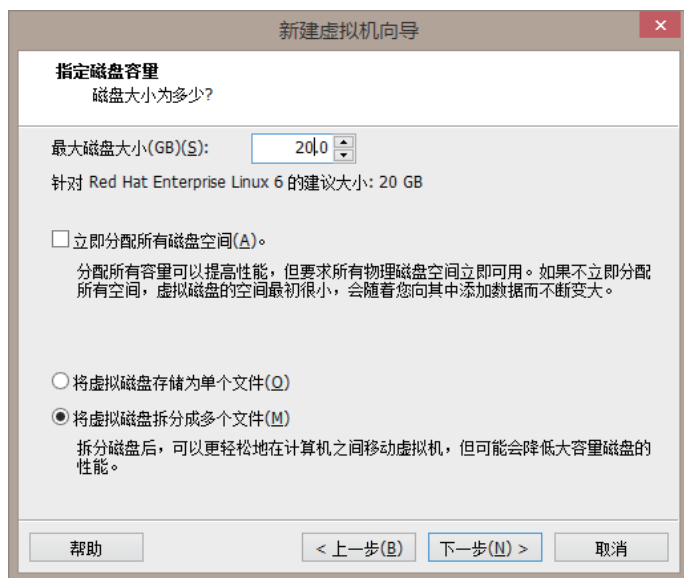
图A-12 选择磁盘类型

**Step 10** 选择磁盘，如图 A-13 所示。



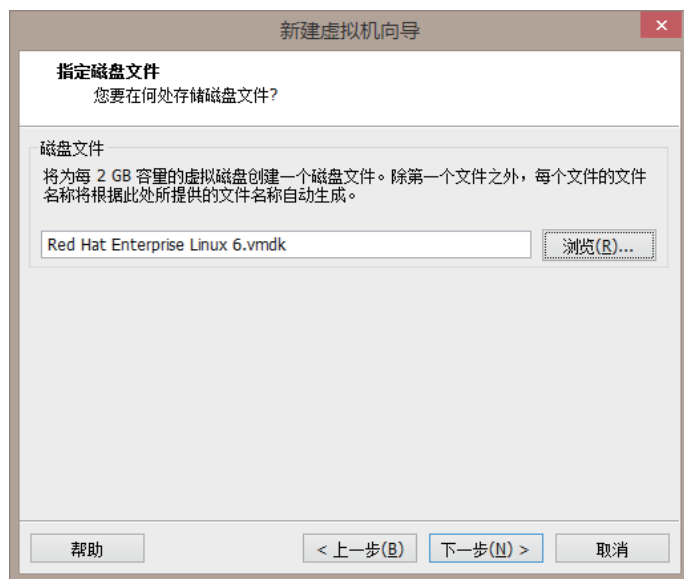
图A-13 选择磁盘

**Step 11** 指定磁盘容量，如图 A-14 所示。



图A-14 指定磁盘容量

**Step 12** 指定磁盘文件，如图 A-15 所示。



图A-15 指定磁盘文件

**Step 13** 完成创建虚拟机，如图 A-16 所示。



图A-16 完成创建虚拟机

虚拟机创建完成后，任何时候仍可以通过虚拟机选项卡界面的“编辑虚拟机设置”修改虚拟硬件和其他配置，如图 A-17 所示。



图A-17 虚拟机选项卡

## 安装客户机操作系统

此时名为“Red Hat Enterprise Linux 6”的虚拟机几乎完全等同于一台真实的物理机，所以可以采用完全与安装系统到 PC 或服务器相同的方式安装需要的系统，包括通过光盘安装、网络安装、硬盘安装、USB 设备安装等。

这里采用光盘镜像方式安装 Red Hat Enterprise Linux 6.5 x86\_64。

步骤如下：

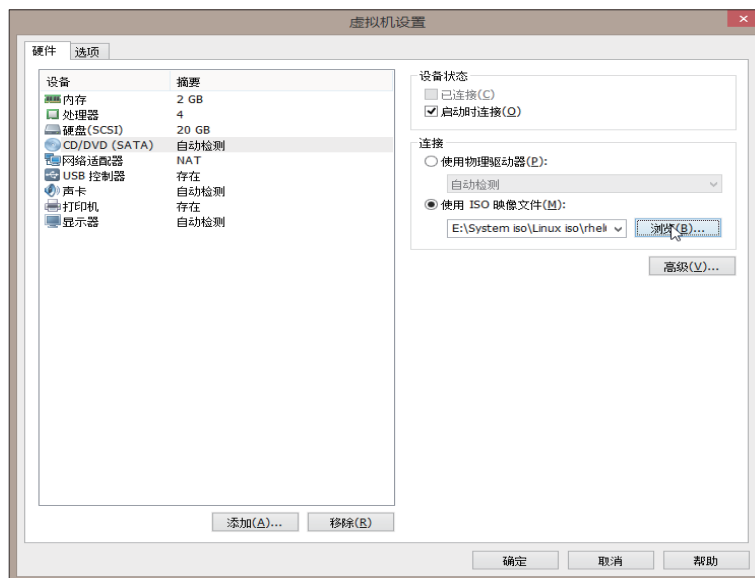
**Step 01** 单击“设备”→“CD/DVD”，弹出“虚拟机设置”对话框，如图 A-18 所示。



图A-18 设置虚拟光驱

**Step 02** 单击“连接”→“使用 ISO 映象文件”，单击“浏览”按钮，选择 ISO 文件路径，如图 A-19 所示。

E:\System iso\Linux iso\rhel6.5\_x32\rhel-server-6.5-i386-dvd.iso

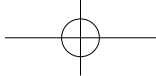


图A-19 使用ISO映象文件

**Step 03** 确定后，单击“开启此虚拟机”，进入系统安装界面（系统安装过程参考第2章），如图 A-20 所示。



图A-20 安装引导界面



## A.5 使用虚拟机

对于安装好的虚拟机，可以进行启动、停止、暂停等操作，也可以创建虚拟机快照用来将虚拟机还原到某个状态，甚至可以基于某个已经安装好的虚拟机通过克隆的方式快速安装。

### A.5.1 虚拟机的启动、停止、暂停等操作

(1) 单击左侧“库”中“我的计算机”虚拟机列表中的一台需要启动的虚拟机，通过以下两种方式启动：

- 菜单“虚拟机”→“电源”→“启动客户机”。
- 虚拟机选项卡→“开启此虚拟机”。

(2) 对于已经安装好客户机操作系统的虚拟机，通过操作系统的关机命令进行关机停止。

(3) 可以通过菜单“虚拟机”→“电源”，对虚拟机进行强制停止、暂停/继续、挂起/恢复等操作。

### A.5.2 创建虚拟机快照

(1) 菜单“虚拟机”→“快照”→“拍摄快照”，创建虚拟机快照（还原点）。

(2) 菜单“虚拟机”→“快照”→“恢复到快照：快照名”，实现从某个快照点还原虚拟机状态。

(3) 菜单“虚拟机”→“快照”→“快照管理器”，管理快照。

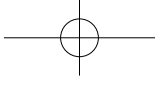
### A.5.3 克隆虚拟机

使用虚拟机克隆功能可以快速创建一个当前虚拟机的副本，克隆出来的虚拟机和原始虚拟机具有相同的硬件属性（唯一性硬件属性不相同，如 MAC）、客户机系统及用户设置。

需要被克隆的虚拟机必须处于停止状态。

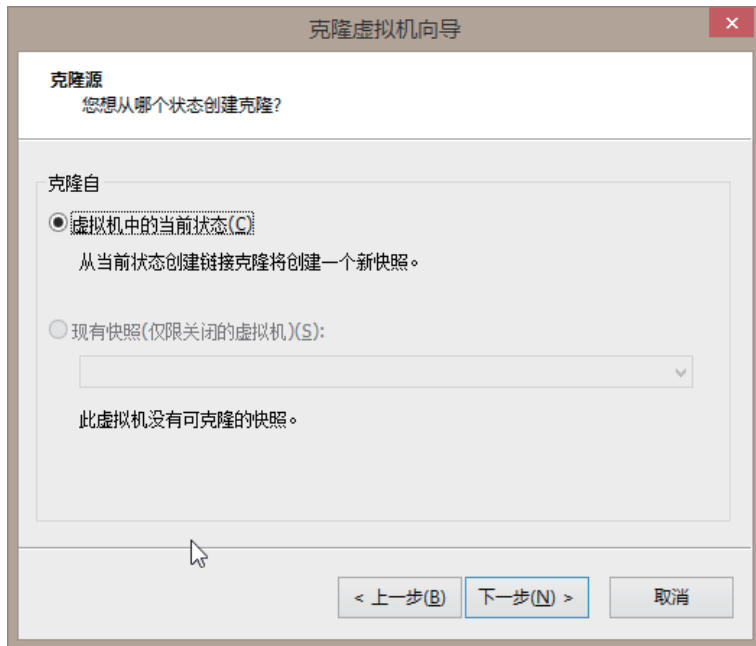
步骤如下：

**Step 01** 单击菜单“虚拟机”→“管理”→“克隆”，弹出虚拟机克隆向导，如图 A-21 所示。

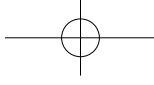


图A-21 克隆虚拟机向导

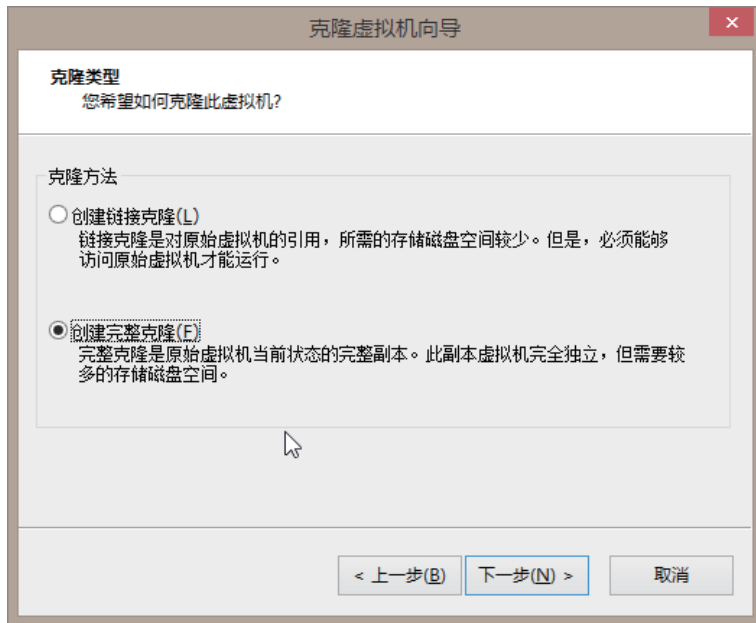
**Step 02** 选择克隆源，如图 A-22 所示。



图A-22 克隆源

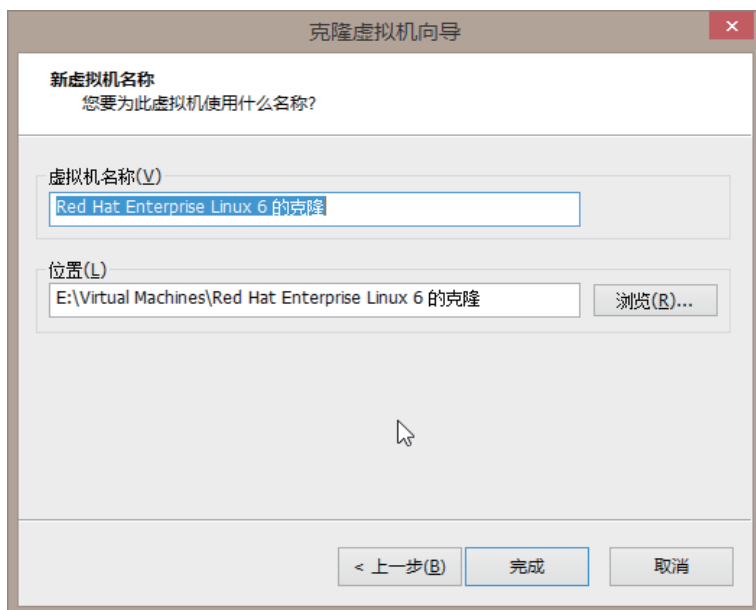


**Step 03** 选择克隆方法，非存储空间不够的情况下，推荐使用“创建完整克隆”，如图 A-23 所示。



图A-23 克隆类型

**Step 04** 指定新虚拟机的名称和虚拟机文件存放的位置，如图 A-24 所示。

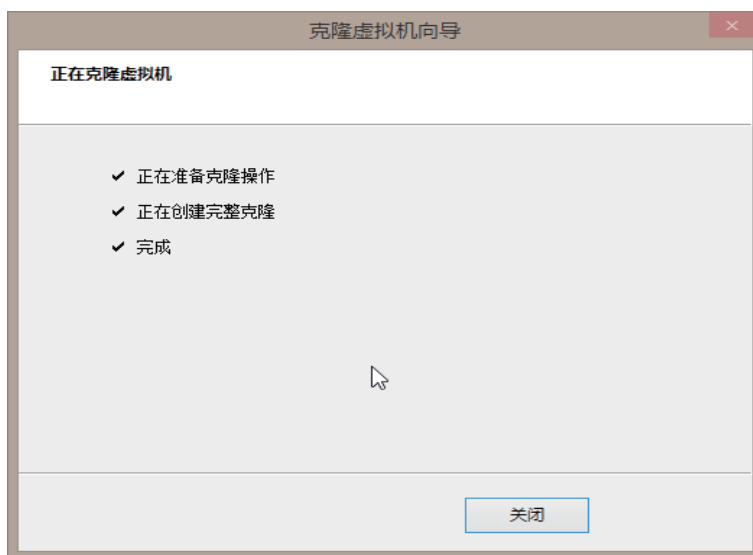


图A-24 新虚拟机名称





**Step 05** 完成虚拟机克隆，如图 A-25 所示。



图A-25 完成虚拟机克隆

## A.5.4 添加或更改虚拟机硬件

在虚拟机使用过程中，可能需要对虚拟机的硬件配置作出调整以满足客户端系统的应用需求，比如增大内存、增加虚拟硬盘、调整网络连接等。

以增加虚拟硬盘和调整网络连接为例。



### NOTE

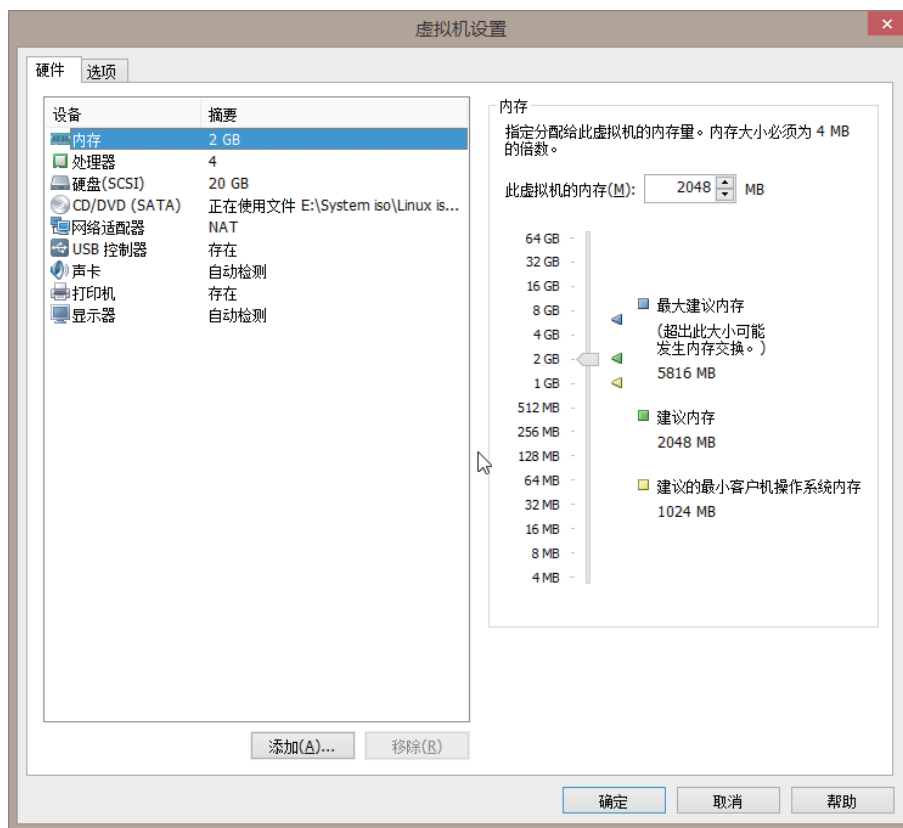
如果客户机操作系统正在运行，在调整虚拟机硬盘设置后可能需要重启客户机系统。

### 1. 增加虚拟硬盘

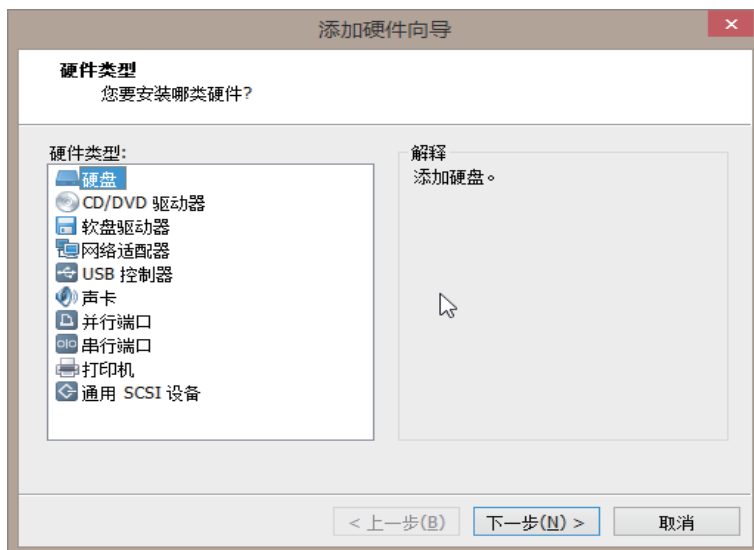
步骤如下：

**Step 01** 在相应的虚拟机选项卡中，单击“编辑虚拟机设置”，弹出“虚拟机设置”对话框，如图 A-26 所示。

**Step 02** 在“硬件”选项卡中单击“添加”按钮，选择要添加的硬件类型为硬盘，如图 A-27 所示。



图A-26 虚拟机设置

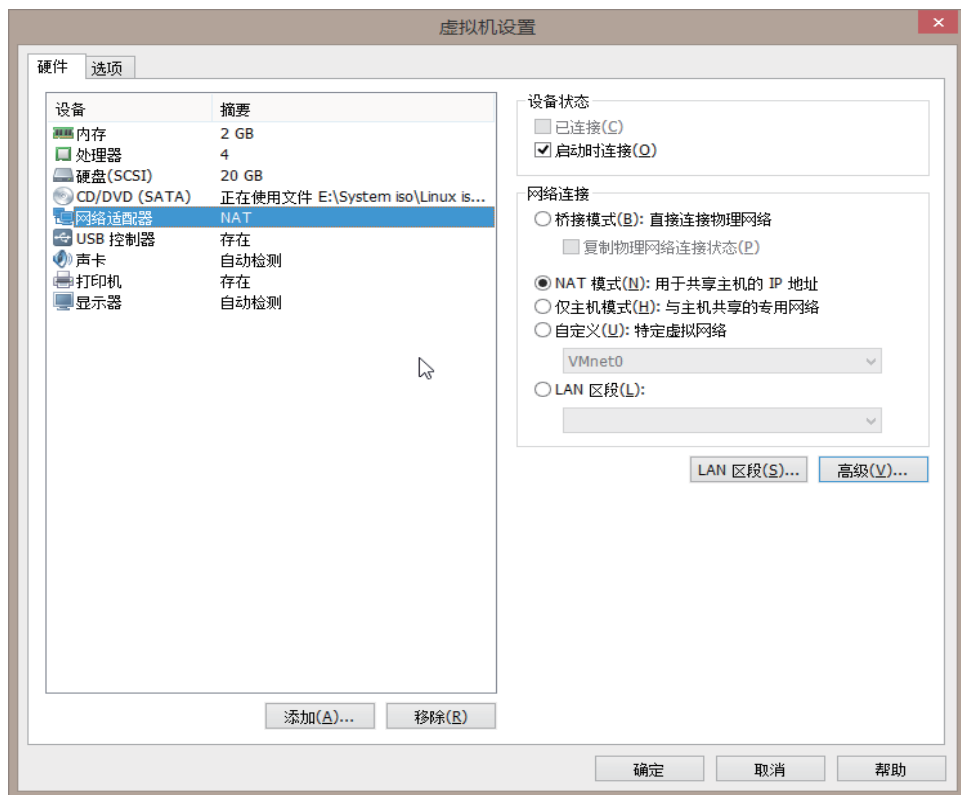


图A-27 硬件类型

**Step 03** 后续定义硬盘属性操作与创建虚拟机步骤相同，不再重复。

## 2. 调整网络连接

在相应的虚拟机选项卡中，单击“编辑虚拟机设置”，在弹出的对话框的“硬件”选项卡中单击选择“网络适配器”，更改需要的网络连接类型，如图 A-28 所示。



图A-28 网络适配器

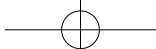
常用的 3 种网络连接类型如下。

### 1) 桥接模式

直接连接物理网络。在这种方式下，虚拟机（Guest）和真实物理机（Host）拥有相同的网络连接属性。比如，物理的 IP 是 192.168.1.101/24，则虚拟机的 IP 是 192.168.1.102/24，相当于在 192.168.1.0/24 网段内增加一台网络主机。虚拟机和宿主物理机能够互相进行网络通信访问；虚拟机和本网段内其他物理机、本宿主机上其他使用桥接模式的虚拟机、其他物理机上使用桥接模式的虚拟机都能够互相进行网络通信访问。该模式常用于测试外部网络访问虚拟机的场景，即虚拟机等同于物理机。

### 2) NAT 模式

用于共享主机的 IP 地址。NAT 即 Network Address Translation（网络地址转换），在这种方式下，虚拟机拥有自己的私有子网地址，并通过真实物理机上的虚拟网络接口 VMnet 连



接物理机网络，如图 A-29 所示。比如，虚拟机的 IP 是 192.168.148.130/24，VMnet8 的 IP 是 192.168.148.1/24。虚拟机和宿主物理机能够互相进行网络通信访问；虚拟机能通过宿主物理机的 NAT 功能访问其他物理机，反之其他物理机不能访问该虚拟机；本宿主主机上使用 NAT 模式的虚拟机之间可以互相访问。该模式常用于不需要外部网络访问虚拟机，仅虚拟机能访问外部网络的场景，虚拟机相当于内网主机。这也是通常默认的方式。



图A-29 Windows网络连接

### 3) 仅主机模式

与主机共享的专用网络。在这种方式下，虚拟机拥有自己的私有子网地址，比如 192.168.24.128/24。虚拟机仅能和宿主物理机、宿主物理机上同样使用仅主机模式的虚拟进行互相网络通信。该模式常用于隔离外部网络，仅虚拟机（或和宿主机）之间进行内部测试的场景。

# Bash常用基础命令用法示例

本附录列出了 Linux Bash 常用的基础命令用法，这些命令属于对于文件操作、系统控制的基础命令，必须要熟练掌握。以下命令的用法在系统中的帮助文档中有更详细的说明描述，在此仅列出部分常用的选项加以说明，帮助初学者更轻松的学习。

## B.1 文件和目录相关的命令

### B.1.1 命令：ls

描述：ls 命令用于查看文件，类似于 DOS 下的 dir 命令。

用法：ls [ 选项 ] 文件或目录名

常用选项：

- -a：列出目录下的所有文件，包括以“.”开头的隐藏文件。
- -l：列出文件的详细信息。
- -d：只显示目录本身。
- -i：列出文件的 i 节点的信息。
- -h：人性化地显示文件的大小。
- -R：递归地显示目录及子目录下的文件。
- -t[-u-S]：以文件被修改的时间 [ 文件上次被访问的时间 / 文件大小 ] 排序。

示例：列出 /boot 目录下的文件，并按文件大小从大到小排列。

```
[demo@365linux ~]$ ls -lhS /boot
总用量 26MB
-rw-----. 1 root root 16M 3月 10 23:57 initramfs-2.6.32-431.el6.i686.
img
```



```
-rw-----. 1 root root 4.0M 3月 5 05:37 initrd-2.6.32-431.el6.i686kdump.  
img  
-rwxr-xr-x. 1 root root 3.9M 11月 11 11:26 vmlinuz-2.6.32-431.el6.i686
```

## B.1.2 命令：mkdir

描述：创建新目录。

用法：mkdir [-pm] 目录名

示例：递归创建目录树。

```
[demo@365linux ~]$ mkdir -p test/a/b/c
```

## B.1.3 命令：touch

描述：创建一个新文件，当被创建的文件存在时，则改变文件的时间戳。

用法：touch [-acmdt] 文件名

常用选项：

- -a：修改文件的访问时间。
- -c：仅用来修改文件的时间，如果目标文件不存在，不会建立新的文件。与 --no-create 的效果相同。
- -m：修改文件的修改时间。
- -d：后面接日期，也可以使用 --date=" 日期或时间 "。
- -t：后面接日期，格式为 [YYMMDDhhmm]。

示例：创建 3 个空文件 a、b、c。

```
[demo@365linux ~]$ touch a b c
```

示例：创建 3 个空文件 file001、file002、file003。

```
[demo@365linux ~]$ touch file{001,002,003}
```

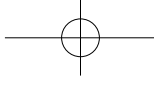
## B.1.4 命令：rmdir

描述：删除一个目录。

用法：rmdir 目录名

示例：删除空目录 test。若目录不为空时，命令执行会报错。

```
[demo@365linux ~]$ rmdir /home/demo/test
```



### B.1.5 命令：rm

描述：删除文件和目录。

用法：rm [-irf] 文件名

常用选项：

- -i：进行删除前必须先确认。
- -r：递归地删除目录及其子目录下的所有文件。
- -f：进行删除操作时不提示。



#### REAL-WORK TIP

rm 命令的 -rf 组合选项在工作中很常用，但使用时要非常谨慎，特别是在使用 root 用户登录工作的情况下。在执行命令前请先确定您要删除的文件和路径不会有任何问题，一旦执行，rm -rf 命令将快速删除目标目录下的一切文件。使用 rm 删除文件没有提示，删除速度快，且删除后很难恢复。

示例：删除目录 test03 和目录下所有的文件，包括子目录，删除时不需要确认。

```
[demo@365linux ~]$ rm -rf /home/demo/test03
```

### B.1.6 命令：cp

描述：复制文件或目录。

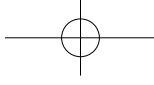
用法：cp [-adpRfiuv] 源文件或目录名 目标文件或目的目录

常用选项：

- -a：相当于 -dpR。
- -d：当复制符号链接时，把目标文件或目录也建立为符号链接，并指向与源文件链接的原始文件或目录。
- -p：保留文件或目录的属性。
- -R：递归地复制目录及目录下的文件和子目录。
- -f：强制复制，覆盖前不提示。
- -i：覆盖前提示确认。
- -u：只有源文件的修改时间较目标文件更新时或目标文件不存在时才会复制。
- -v：显示指令执行过程。

示例：复制多个文件到 /tmp。

```
[demo@365linux ~]$ cp /etc/fstab /etc/man.config /tmp/
```



示例：备份 /etc 目录下的所有文件和目录（包括子目录）到 backup，目的目录 backup 必须存在。

```
[demo@365linux ~]$ cp -a /etc/* /home/demo/backup
```

示例：复制 /etc/passwd 到 /tmp 并改名为 passwd.bak。

```
[demo@365linux ~]$ cp /etc/passwd /tmp/passwd.bak
```

### B.1.7 命令：mv

描述：移动或重命名文件或目录。

用法：mv [-ifuv] 源文件或目录名 目标文件或目的目录

示例：将 passwd.bak 重命名为 passwd.bak.mv。

```
[demo@365linux ~]$ mv /tmp/passwd.bak /tmp/passwd.bak.mv
```

### B.1.8 命令：file

描述：查看某个文件的基本信息。

用法：file 文件名

示例：使用 file 命令查看 /bin/ls 的文件信息。

```
[demo@365linux ~]$ file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.18, stripped
```

### B.1.9 命令：ln

描述：为源文件创建一个链接文件（软链接或硬链接）。

用法：ln [-s] 源对象 目的对象

示例：为 test.txt 创建一个硬连接 test.txt.ln。

```
[demo@365linux ~]$ ln test.txt test.txt.ln
```

示例：为 test.txt 创建一个软链接。

```
[demo@365linux ~]$ ln -s test.txt test.txt.lns
```

示例：查看和比较硬链接与软链接的区别。

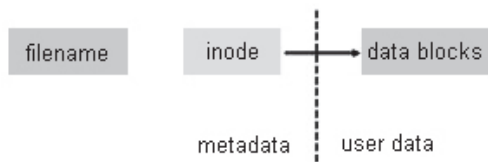
```
[demo@365linux ~]$ ll -li test.txt*
785424 -rw-rw-r--. 2 demo demo 0 3月 18 16:23 test.txt
785424 -rw-rw-r--. 2 demo demo 0 3月 18 16:23 test.txt.ln
785521 lrwxrwxrwx. 1 demo demo 8 3月 18 16:24 test.txt.lns -> test.txt
```

Linux 文件将文件存储设计分成两部分，即文件元数据（metadata）和用户数据（user data）。用户数据也就是文件数据块（data block），数据块是记录文件真实内容的地方；而元



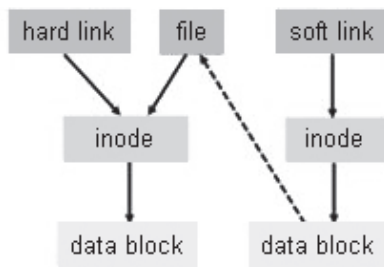


数据则是文件的附加属性，如文件大小、创建时间、拥有者、权限等信息。在 Linux 系统中，元数中的 inode 号(inode 是文件元数据的一部分但其并不包含文件名,inode 号即索引节点号)才是文件的唯一标识而非文件名。文件名仅是为了方便人们的记忆和使用，系统或程序通过 inode 号寻找正确的文件数据块。文件结构如图 B-1 所示。



图B-1 文件结构

为解决文件的共享使用，Linux 系统引入了两种链接：硬链接(hard link)与软链接(又称符号链接，即 soft link 或 symbolic link)，如图 B-2 所示。链接为 Linux 系统解决了文件的共享使用，还带来了隐藏文件路径、增加权限安全及节省存储等好处。若一个 inode 号对应多个文件名，则称这些文件为硬链接。换言之，硬链接就是同一个文件使用了多个别名。



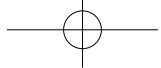
图B-2 硬链接和软链接

由于硬链接是有着相同 inode 号仅文件名不同的文件，因此硬链接存在以下几点特性：

- 文件有相同的 inode 及 data block。
- 只能对已存在的文件进行创建。
- 不能交叉文件系统进行硬链接的创建。
- 不能对目录进行创建，只可对文件进行创建。
- 删除一个硬链接文件并不影响其他有相同 inode 号的文件。

软链接与硬链接不同，若文件用户数据块中存放的内容是另一文件的路径名的指向，则该文件就是软链接。软链接就是一个普通文件，只是数据块内容有点特殊。软链接有着自己的 inode 号及用户数据块，因此软链接的创建与使用没有类似硬链接的诸多限制。

- 软链接有自己的文件属性及权限等。
- 可对不存在的文件或目录创建软链接。



- 软链接可交叉文件系统。
- 软链接可对文件或目录进行创建。
- 创建软链接时，链接计数 `i_nlink` 不会增加。
- 删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软链接被称为死链接（Dangling Link）。若被指向路径文件被重新创建，死链接可恢复为正常的软链接。

## B.2 显示和查看相关的命令

### B.2.1 命令：cat

描述：显示文件的内容。

用法：cat [-AETvnbs] 文件名

常用选项：

- `-A`：相当于 `-Etv`，可列出一些特殊字符。
- `-E`：将结尾的换行符 `$` 显示出来。
- `-T`：将 `[tab]` 间隔以 `^I` 显示出来。
- `-v`：列出一些看不出来的特殊字符。
- `-n`：显示行号。
- `-b`：和 `-n` 相似，只是不对空白行编号。
- `-s`：当遇到连续两行以上的空白行时，就替换为一行。

示例：显示文件 `passwd` 的内容。

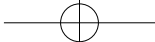
```
[demo@365linux ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
```

### B.2.2 命令：more

描述：分页显示文件的内容。

用法：more 文件名

more 命令的功能键如表 B-1 所示。



表B-1 页码导航控制

按 键	功 能
空格键	向下翻一页
Enter 键	向下翻一行
/ 字符串	向下搜索“字符串”
n 键	继续向下查找
q 键	退出 more 显示

示例：使用 more 命令查看 man.config 文件。

```
[demo@365linux ~]$ more /etc/man.config
```

使用空格键向下翻页，使用 Enter 键向下翻一行。输入 /，光标会移动到屏幕下方，输入要查询的字符串，按 Enter 键，开始向下搜索匹配的内容。要重复搜索同一字符串，按 n 键。最后按 q 键退出。

### B.2.3 命令：less

描述：分页显示文件的内容，支持向上翻页。

用法：less 文件名

功能键与 more 相似，支持 [pagedown][pageup]?N。

示例：使用 less 命令查看 man.config 文件。

```
[demo@365linux ~]$ less /etc/man.config
```

more 命令不能实现向上翻页和向上搜索，less 命令可以实现该功能。除了具备 more 的功能外，less 还支持使用 [pagedown] [pageup] 向下向上翻页，使用 ? 向上搜索字符串，使用 N 反向查询。

### B.2.4 命令：head

描述：显示文件的前  $n$  行，默认显示前 10 行。

用法：head [-n] 文件名

示例：查看指定文件的前 10 行。

```
[demo@365linux ~]$ head /etc/passwd
```

示例：查看指定文件的前 5 行的两种用法。

```
[demo@365linux ~]$ head -n 5 /etc/passwd
```

```
[demo@365linux ~]$ head -5 /etc/passwd
```



## B.2.5 命令：tail

描述：显示文件的后  $n$  行，默认显示后 10 行。

用法：tail [-nf] 文件名

示例：显示日志文件 /var/log/messages 的后 10 行。

```
[root@365linux ~]# tail /var/log/messages
```

示例：显示日志文件 /var/log/messages 的后 20 行。

```
[root@365linux ~]# tail -n 20 /var/log/messages
```

示例：动态显示日志文件 /var/log/messages 的后 20 行。当文件内容有变化时，此命令动态显示；要退出显示，按 Ctrl+C 组合键。

```
[root@365linux ~]# tail -f /var/log/messages
```

## B.3 查找相关的命令

### B.3.1 命令：find

描述：从指定的起始目录开始，递归地查找其各个子目录，查找满足条件的文件并对之采取相关的操作。

用法：find 起始目录 查找条件 执行操作

查找条件：-name -iname -user -group -uid -gid -empty -size -path -perm -mount -nouser -nogroup -type -xtype -amin -atime -cmin -ctime -mmin -mtime -newer

执行操作：-exec -ok -ls -print

逻辑运算符：-a(and) -o(or) !(not)

示例：查找 /etc 目录下（包含子目录）文件名以 .conf 结尾的文件。使用 -iname，不区分大小写。

```
[root@365linux ~]# find /etc -name "*.conf"
```

示例：在多个目录下进行查找。

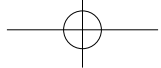
```
[root@365linux ~]# find /usr /home /tmp -name "*.jar"
```

示例：查找当前目录（包含子目录）中拥有者是 root 的文件。

```
[root@365linux ~]# find . -user root
```

示例：查找 /home（包含子目录）目录中在 24 小时内被访问过的文件。

```
[root@365linux ~]# find /home -atime 0
```



示例：查找 /home 目录（包含子目录）中拥有者和用户组不存在的文件，并执行删除操作。

```
[root@365linux ~]# find /home -nouser -a -nogroup -exec rm -rf {} \;
```

示例：同上，但执行操作时需要一个一个确认。

```
[root@365linux ~]# find /home -nouser -a -nogroup -ok -rm -rf {} \;
```

示例：查找 test 目录中大小为零的文件并把它们移动到 /tmp/zerobyte 目录中。

```
[root@365linux ~]# find test -type f -size 0 -exec mv {} /tmp/zerobyte \;
```

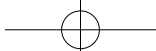
示例：查找当前目录中权限为 777 的文件，并列出详细列表。

```
[root@365linux ~]# find . -type f -perm 777 -ls
```

find 命令常用查找条件如表 B-2 所示。

表B-2 find命令常用查找条件

查找条件	说 明
-name filename	查找文件名为 filename 的所有文件，可用通配符？ *[]
-user username	查找拥有者为 username 的所有文件
-gourp gourpname	查找所属组为 gourpname 的所有文件
-uid n	查找用户 UID 为 n 的所有文件
-gid n	查找组 GID 为 n 的所有文件
-empty	查找空文件
-size [+/-] SIZE [bcwkMG]	查找大小为（大于 / 小于）SIZE 的文件，单位可以是 [bcwkMG]
-perm [-/] mode	查找权限值为（含有 / 至少）mode 的文件
-nouser	查找拥有者不存在的文件
-nogroup	查找所属组不存在的文件
-type [bcdpfls]	查找类型为 [b:块设备文件 c:字符设备文件 d:目录 p:管道文件 f:普通文件 l:链接文件 s:Socket 文件]
-atime n	查找在 n 天前的“一天之内”被访问过的文件（mtime；ctime）。 关于 n 的解释如下。 •n: n 天前的“一天之内”。例如，0 指从现在起之前的一天以内，1 指一天之前的一天以内。 •+n: n+1 天之前的所有。例如，+0 指一天以前的所有，+1 指两天之前的所有。 •-n: n 天以内的所有。例如，-1 指一天以内的所有，-2 指两天以内的所有
-amin n	查找在 n 分钟前被访问的文件。注意：-n 表示文件被访问时间距现在 n 分钟以内，+n 表示文件被访问时间距现在 n 分钟以前
-mmin n	n 分钟前文件内容被修改
-cmin n	n 分钟前文件属性被修改（包括创建到文件的符号链接、更改文件权限或移动了文件等）
-newer file	查找比 file 新的文件
-mount 或 -xdev	在查找文件时不跨越文件系统
-fstype	明确告知 find 在哪个文件系统中查找
-depth	在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找
-prune	使用这一选项可以使 find 命令不在当前指定的目录中查找，如果同时使用 -depth 选项，那么 -prune 将被 find 命令忽略
-mount	在查找文件时不跨越文件系统 mount 点
-follow	如果 find 命令遇到符号链接文件，就跟踪至链接所指向的文件



续表

查找条件	说 明
-cpio	对匹配的文件使用 cpio 命令，将这些文件备份到磁带设备中
-maxdepth 和 -mindepth	允许指定希望 find 搜索深入到目录树的哪一级别

### find 命令执行操作技巧：

-exec 选项处理匹配到的文件时，find 命令会将所有匹配到的文件一起传递给 exec 来执行，如果参数过多，有可能产生“参数列溢出”的错误。另外，在有些系统中，使用 -exec 选项会为处理每一个匹配的文件而发起一个相应的进程，如果进程过多会使系统性能下降。为避免上述问题，我们可以用 xargs 命令来配合 find 命令一起使用。除此之外，xargs 在某些应用中将结果作为参数引用更加灵活（xargs 命令对匹配结果分批处理并只产生一个进程）。

示例：比较以下两个命令的执行效率。

```
[root@365linux ~]# find / -type f -exec ls -l {} \;  
[root@365linux ~]# find / -type f |xargs ls -l
```

## B.3.2 命令：locate(updatedb)

描述：快速从系统数据库中查找指定的内容。

示例：将完整文件名（含路径）中包含 file001 的文件显示出来。

```
[root@365linux ~]# locate file001
```

示例：使用 -r 参数可以使 locate 支持正则表达式。

```
[root@365linux ~]# locate -r '\<passwd\>$'
```

locate 查找数据是根据已建立的数据库 /var/lib/mlocate/mlocatedb 里面的数据搜索得到的。RHEL 默认每天更新一次数据内容，手动更新数据库使用 updatedb 命令。

## B.3.3 命令：which

描述：查询命令所在的位置。

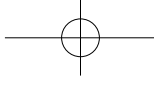
用法：which 命令名称

示例：显示 ls 命令的位置及别名相关信息。

```
[demo@365linux ~]$ which ls
```

## B.3.4 命令：whereis

描述：查询指定文件的源、二进制文件和手册等。



示例：显示 ls 命令的位置及帮助手册。

```
[demo@365linux ~]$ whereis ls
```

示例：显示 /etc/passwd 文件相关的命令及帮助手册。

```
[demo@365linux ~]$ whereis /etc/passwd
```

### B.3.5 命令：grep

描述：在指定的文件或标准输出、标准输入内查找满足条件的内容，起过滤作用，支持正则表达式。

用法：grep [-invRE] 文件名

示例：显示 /etc/passwd 文件中含有 root 的行，并在行首加上所在行号。

```
[demo@365linux ~]$ grep -n root /etc/passwd
```

示例：显示 /etc/passwd 文件中不包含 root 的行。

```
[demo@365linux ~]$ grep -v root /etc/passwd
```

示例：显示 /etc 目录中（-R：包括子目录）的所有文件中包含以 yes 结尾的行，查找时忽略大小写。

```
[demo@365linux ~]$ grep -iR 'yes$' /etc/*
```

示例：-q 选项表示安静模式，grep 命令不输出结果，可以通过命令返回值“\$?”判断，常用于 Shell 脚本中。

```
[demo@365linux ~]$ grep -q root /etc/passwd
[demo@365linux ~]$ echo $?
0
```

egrep 等于 grep -E；fgrep 等于 grep -F。

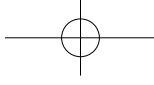
## B.4 查看系统信息相关的命令

### B.4.1 命令：who (w, whoami, users)

描述：显示系统中有哪些用户登录及其来源。

示例：查看当前系统登录的用户。

```
[demo@365linux ~]$ who
root      tty2          2014-03-18 18:15
root      pts/0         2014-03-18 15:42 (192.168.148.1)
demo      tty1          2014-03-18 18:15 (:0)
demo      pts/1         2014-03-18 18:16 (:0.0)
```



说明：第一列是用户名；第二列表示从哪个终端登录，`tty1` 表示虚拟控制台 1，`:0` 指 X 服务器本身，`pts/1`（第三列有 `:0.0`）表示在本机的 X 服务器上打开的第一个虚拟终端，`pts/0`（第三列有 IP）表示从远程终端登录；第三列是用户登录的时间和来源。

## B.4.2 一系列与登录用户相关的命令

### 1. 命令：`w`

描述：当前登录到本机的用户及运行的程序。

### 2. 命令：`whoami`

描述：当前在本地登录到本机的用户。

### 3. 命令：`finger`

描述：显示用户的资料。

### 4. 命令：`id`

描述：显示用户的身份。

### 5. 命令：`users`

描述：显示当前系统登录的用户。

### 6. 命令：`write`

描述：给当前联机的用户发消息。

### 7. 命令：`wall`

描述：给所有登录在别处的用户广播消息。

### 8. 命令：`last`

描述：查看用户的登录日志。

### 9. 命令：`lastlog`

描述：查看每个用户最后登录的情况。

## B.4.3 命令：`date`

描述：显示或设置系统的日期和时间。

用法：`date [-dsu] [+FORMAT] [MMDDhhmm][[CC]YY][.ss]`





常用选项：

- **-d datestr**：显示 datestr 中所设定的日期或时间（非系统时间）。
- **-s datestr**：将系统日期或时间设为 datestr 中所设定的日期或时间。
- **-u**：显示目前的格林尼治时间。

使用 **hwclock** 命令同步系统时间和硬件时间。

## 1. 显示系统时间

示例：以各种不同的格式显示系统当前时间。

```
[demo@365linux ~]$ date
2014 年 03 月 18 日 星期二 18:27:48 CST
```

```
[demo@365linux ~]$ date '+%Y-%m-%d %H:%M:%S'
2014-03-18 18:28:54
```

```
[demo@365linux ~]$ date "+%F %T"
2014-03-18 18:29:17
```

```
[demo@365linux ~]$ date -d "2 days ago"
2014 年 03 月 16 日 星期日 18:29:51 CST
```

```
[demo@365linux ~]$ date -d "1970-1-1 15872 days" "+%F"
2013-06-16
```

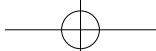
**date** 显示当前系统时间，可以指定时间显示格式。格式如表 B-3 和表 B-4 所示。

表B-3 时间格式

格 式	含 义	格 式	含 义
%	印出 % 号	%S	秒 (00..60)
%n	下一行	%p	显示本地 AM 或 PM
%t	跳格	%r	直接显示时间 (12 小时制, 格式为 hh:mm:ss[AP]M)
%H	小时 (00..23)	%s	从 1970 年 1 月 1 日 00:00:00UTC 到当前为止的秒数
%I	小时 (01..12)	%Z	显示时区
%l	小时 (1..12)	%T	直接显示时间 (24 小时制)
%M	分钟 (00..59)	%X	相当于 (%H : %M : %S)

表B-4 日期格式

格 式	含 义	格 式	含 义
%a	星期几 (Sun..Sat)	%h	同 %b
%A	星期几 (Sunday..Saturday)	%j	一年中的第几天 (001..366)
%b	月份 (Jan..Dec)	%m	月份 (01..12)



续表

格 式	含 义	格 式	含 义
%W	一年中的第几周 (00..53) (以 Monday 为一周的第一天情形)	%U	一年中的第几周 (00..53) (以 Sunday 为一周的第一天情形)
%c	直接显示日期与时间	%w	一周中的第几天 (0..6)
%d	日 (01..31)	%B	月份 (January..December)
%D	直接显示日期 (mm/dd/yy)	%x	直接显示日期 (mm/dd/yy)
%y	年份的最后两位数字 (00..99)	%Y	完整年份 (0000..9999)

更多时间日期的格式可以通过 `date` 的帮助文档来进行查看。

## 2. 设置系统时间

要设定时间,不使用“+”号,而时间格式为 `MMDDhhmm[[CC]YY][.ss]`,其中 `MM` 为月份, `DD` 为日, `hh` 为小时, `mm` 为分钟, `CC` 为年份前两位数字, `YY` 为年份后两位数字, `ss` 为秒数。

示例: 设置系统时间可用的各种时间格式。

```
[root@365linux ~]# date 090811272010.45
[root@365linux ~]# date -s "2 days ago"
[root@365linux ~]# date -s 20100708
[root@365linux ~]# date -s 1123
[root@365linux ~]# date -s '2010/07/08'
[root@365linux ~]# date -s '11:25:34'
[root@365linux ~]# date -s '20100908 1127'
```

## B.4.4 命令: hwclock

描述: 查询和设置硬件时钟。

用法: `hwclock [function] [options...]`

示例: 查看硬件时钟时间。

```
[root@365linux ~]# hwclock --show
```

示例: 将系统时间同步到硬件时钟。

```
[root@365linux ~]# hwclock --systohc
```

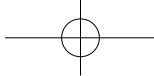
示例: 将硬件时钟时间同步到系统时钟。

```
[root@365linux ~]# hwclock --hctosys
```

## B.4.5 命令: cal

描述: 显示日历。

用法: `cal [-m]y [month [year]]`



## B.5 系统管理相关命令

### B.5.1 命令：free

描述：查看当前系统内存的使用情况。

用法：free [-bkmgltsc]

常用选项：

- -b,-k,-m,-g：以字节、KB、MB、GB 为单位显示大小。
- -s n：持续显示 free 结果，间隔时间 n 秒；可与 -c n 一起使用，表示持续显示 n 次。

示例：查看系统内存使用量。

```
[demo@365linux ~]$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	2022	959	1062	0	108	547
-/+ buffers/cache:		304	1717			
Swap:	4095	0	4095			

- Mem：表示物理内存。
- -/+ buffers/cache：表示缓存统计。
- Swap：表示交换分区。
- total：总量。
- used：使用量（包括 buffers 与 cache）。
- shared：共享内存。
- buffers：系统分配但未被使用的 buffers 数量（与块设备有关的高速缓冲区）。
- cached：系统分配但未被使用的 cached 数量（与文件 inode 有关高速缓冲区）。

从应用程序的角度来说，可用内存 = 系统 free + buffers + cached。

### B.5.2 命令：df

描述：查看磁盘用量。

用法：df [-ahTxi] [file]

常用选项：

- -a：显示包含虚拟文件系统，默认不显示。
- -h：以人性化的格式显示单位。
- -i：显示 inode 的使用信息。
- -l：只显示本地文件系统。



- **-t** : 指定显示的文件系统类型。
- **-T** : 显示每个文件系统类型。
- **-x** : 指定不显示的文件系统类型。

**示例** : 查看当前系统磁盘空间的使用量。

```
[demo@365linux ~]$ df -Th
Filesystem      Type      Size    Used Avail Use% Mounted on
/dev/sda3       ext4      16G    3.2G   12G   22% /
tmpfs           tmpfs    1012M    84K 1012M    1% /dev/shm
/dev/sda1       ext4      485M    37M   423M    8% /boot
```

### B.5.3 命令: du

**描述** : 显示文件或目录的大小。

**用法** : `du [-sh] [ 文件或目录名 ]`

**常用选项** :

- **-s** : 只显示每个参数的总量信息。
- **-h** : 以人性化的格式显示单位。
- **-S** : 单独统计每个目录的大小, 不包括子目录。
- **-L** : 不统计符号链接的文件。
- **-x** : 跳过在不同的文件系统上的目录。

**示例** : 查看指定目录包含内容占用磁盘空间大小。

```
[demo@365linux ~]$ du -sh /home/demo
15M /home/demo
```

### B.5.4 命令: shutdown

**相关命令** : `poweroff halt reboot`

**描述** : 关机或重启命令。

**用法** : `shutdown [-rkhc] time [ 警告信息 ]`

**示例** : 马上执行关机操作。

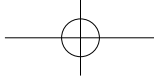
```
[root@365linux ~]# shutdown -h now
```

**示例** : 5 分钟以后执行关机操作。

```
[root@365linux ~]# shutdown -h 5
```

**示例** : 并不是真正关机, 只是发出警告信息。

```
[root@365linux ~]# shutdown -k now "The system is going down."
```



示例：执行重新启动的操作。

```
[root@365linux ~]# shutdown -r now
```

示例：取消一个已经运行的关机命令。

```
[root@365linux ~]# shutdown -c
```

## B.5.5 命令：init (telinit)

描述：改变系统的运行级别。

RHEL 系统默认设置 7 个运行级别，分别如下。

- 0：关机。
- 1：单用户模式。
- 2：不带网络的多用户模式。
- 3：全功能的多用户模式，默认不进入 X Window 模式。
- 4：备用。
- 5：默认进入 X Window 模式。
- 6：重启。

用法：init [0123456]

示例：使用 init 命令切换运行级别到 0 实现关机操作。

```
[root@365linux ~]# init 0
```

shutdown -h now 命令向所有连接在系统上的用户发送关机的消息，然后调用 init 0，而 init 0 关闭所有的服务，最后调用 halt -p -f 关闭电源。

关机的 3 个命令是有层次的。需要注意的是：如果直接输入 halt 命令将调用 shutdown 命令；但是如果输入 halt -p -f，则直接关闭电源。

shutdown -r now 命令向所有连接在系统上的用户发送重启的消息，然后调用 init 6，而 init 6 将关闭所有的服务，最后调用 reboot。

实际上 reboot 是 halt 命令，但是这时 halt 命令会让系统重新启动。

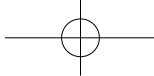
## B.6 打包压缩相关的命令

### B.6.1 命令：gzip

相关命令：zcat zless zdiff gunzip

描述：对文件进行压缩，文件压缩后自动加上 .gz 的扩展名。

用法：gzip [-cdl19] 文件名



示例：压缩文件 test.txt，生成 test.txt.gz。

```
[demo@365linux ~]$ gzip test.txt
```

示例：不解压，查看压缩文件的内容。

```
[demo@365linux ~]$ zcat test.txt.gz
```

示例：将压缩文件解压。

```
[demo@365linux ~]$ gzip -d test.txt.gz
```

或者：

```
[demo@365linux ~]$ gunzip test.txt.gz
```

示例：指定压缩比压缩文件，并保留原文件。压缩等级：1，压缩速度最快，压缩比最差；9，压缩速度最慢，压缩比最好；默认是 6。

```
[demo@365linux ~]$ gzip -9 -c test.txt >test.txt.gz
```

## B.6.2 命令：bzip2

相关命令：bzipcat bzless bzdiff bunzip

## B.6.3 命令：compress

相关命令：uncompress

bzip2 与 gzip 的用法相同，但比 gzip 的压缩效率更高。

compress 是传统的 UNIX 压缩工具，压缩文件以 .Z 为扩展名，使用 uncompress 解压。

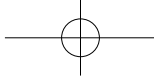
## B.6.4 命令：tar

描述：对多个文件或目录进行打包（解包），默认不压缩。

用法：tar [-cCxtzjvfpPNAdru] [--delete] [--exclude] 文件或目录名

常用选项：

- -N：DATE 比 DATE 新就打包。
- -A：合并 tar 包。
- -d：比较 tar 包内容和文件系统内容的差别。
- -r：向 tar 包内追加文件。
- -u：打包比 tar 包内新的文件。
- -P：压缩时使用绝对路径（默认相对）。
- --delete：删除 tar 包内的文件。
- --exclude：打包时除开某个文件或目录。



示例：将 /etc/ 目录内容打包到 /tmp/etc.tar。

```
[root@365linux ~]# tar -cvf /tmp/etc.tar /etc
```

示例：-z 是指将 /etc 目录内容打包的同时使用 gzip 进行压缩；-j 是指使用 bzip2 进行压缩。

```
[root@365linux ~]# tar -zcvf /tmp/etc.tar.gz /etc
```

示例：不解包，仅查看 tar 包的内容。

```
[root@365linux ~]# tar -tvf /tmp/etc.tar
```

示例：解开 tar 包到当前目录。-z 是指解开 .gz 的压缩包。当不指定解压参数 -z -j 时，tar 命令自动选择可用的参数并调用相关功能。

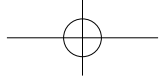
```
[root@365linux ~]# tar -zxvf /tmp/etc.tar.gz
```

示例：解开使用 bzip2 压缩过的 tar 包到指定目录。-C 是指定解压输出目录。

```
[root@365linux ~]# tar -xvf /tmp/etc.tar.bz2 -C /tmp/test
```

示例：-p 参数是指压缩时保留源文件的权限。

```
[root@365linux ~]# tar -zcvpf /tmp/etc.tar.gz /etc
```



## 参考文献

- [1] Redhat 公司官网产品文档：<https://access.redhat.com/documentation/zh-CN/index.html>.
- [2] (美) Michael Jang 著. 吴文国, 李增民, 曲伟 译. RHCSA/RHCE Red Hat Linux 认证学习指南 (第 6 版): EX200 & EX300. 北京: 清华大学出版社, 2013.
- [3] (美) Michael Jang 著. 靳晓辉 译. RHCSA/RHCE Red Hat Linux 认证模拟考试解析与虚拟机 (Exams EX200 & EX300). 北京: 清华大学出版社, 2014.